

Biochemistry Virtual Lab Software

DESIGN DOCUMENT

Team 12

Client: Stone Chen

Faculty Advisor: Simanta Mitra

Peter Bancks

Jacob Christopherson

Enzo Ciccarelli-Asta

Steven Dirth

Romain Ndoutoume

Brody Sunsten

sdmay21-12@iastate.edu

sdmay21-12.sd.ece.iastate.edu

Revised: 11/12/2020

Executive Summary

Development Standards & Practices Used

- Frequent Meetings
- SCRUM
- Standard Naming Conventions
- Unit Testing
 - Mocking
 - CI/CD
- Version Control
 - Git Feature Branch Workflow
 - Frequent & Detailed Commits
 - Task Tracking

Summary of Requirements

- Web Based
 - Web hosting
 - Backend optional
- Interactive 3D Environment & Objects
 - React/Three.js
- Around 7 Modules

Applicable Courses from Iowa State University Curriculum

- BBMB 102
- COM S 309
- COM S 363
- COM S 336
- COM S 339
- SE 319

New Skills/Knowledge acquired that was not taught in courses

- React
- 3d Modeling

Table of Contents

1	Introduction	4
1.1	Acknowledgement	4
1.2	Problem and Project Statement	4
1.3	Operational Environment	4
1.4	Requirements	4
1.5	Intended Users and Uses	5
1.6	Assumptions and Limitations	5
1.7	Expected End Product and Deliverables	5
2	Project Plan	6
2.1	Task Decomposition	6
2.2	Risks And Risk Management/Mitigation	7
2.3	Project Proposed Milestones, Metrics, and Evaluation Criteria	7
2.4	Project Timeline/Schedule	7
2.5	Project Tracking Procedures	8
2.6	Personnel Effort Requirements	8
2.7	Other Resource Requirements	8
2.8	Financial Requirements	9
3	Design	9
3.1	Previous Work And Literature	9
3.2	Design Thinking	9
3.3	Proposed Design	10
3.4	Technology Considerations	10
3.5	Design Analysis	10
3.6	Development Process	11
3.7	Design Plan	11
4	Testing	12
4.1	Unit Testing	12
4.2	Interface Testing	12
4.3	Acceptance Testing	13
4.4	Results	13
5	Implementation	13

6 Closing Material	14
6.1 Conclusion	14
6.2 References	14
6.3 Appendices	14

List of figures/tables/symbols/definitions (This should be the similar to the project plan) (N/A at the moment)

Mesh: The 3D representation of an object that is then displayed on a canvas.

Canvas: A window that allowed for the displaying of 3D object for the lab modules.

1 Introduction

1.1 ACKNOWLEDGEMENT

Professor Chen provided images of various lab equipment as well as the scripts used to create the lab modules for this product. He and Professor Mitra were also involved in forming goals for this project and providing an example project in a 2D environment rather than a 3D one, and a website for the product to be hosted on.

1.2 PROBLEM AND PROJECT STATEMENT

Biochemistry lab courses are essential, inspiring, albeit traditionally difficult components of the curriculum for many biology-related majors. The main barriers to learning biochemical experiments usually come from the abstract nature of biomolecules and biochemical processes and the relatively high demand for equipment, reagents, and hands-on skills. As a result, students often experience frustrations in biochemistry labs due to the stressful pace or lack of understanding of key concepts, which can quench their learning interests.

Our goal is to develop a suite of web-based or app-based virtual lab software, which we expect to greatly enhance the learning experience and outcomes of the biochemistry laboratory course, BBMB102 "Introduction to Biochemistry Laboratory". This software will not only promote student engagement by translating abstract biochemical concepts into vivid animations, but also enhance active learning by allowing students to play with experimental parameters during simulation.

1.3 OPERATIONAL ENVIRONMENT

As the product of this project is a simulation of a chemistry lab, there is no operational environment with physical hazards outside of normal software concerns.

1.4 REQUIREMENTS

Make progressive website with React

Functional Requirements

- 3D environment: User can see and interact with a 3d virtual lab, including objects, tables, PPE, and a notebook.
- Object Interactions: Objects should act according to the module/experiment script to demonstrate interactions. Objects that are not related to the current step will not be interactable.
- Notebook: User should be able to record data necessary for the experiment, graph the data they find, and read instructions and feedback about the experiment.
- Scripted Events: Upon the completion of specified actions, a specific event or reaction will occur, and the experiment will either progress or end.
- User Progress: The user should be able to sign in using Okta SSO. If a user completes a lab, their progress should be saved.

UI requirements

- Must function on all major web browsers

- UI should direct the user to experiments as intended
- Scripts within modules should be error-free and linked to scripted events
- User can interact with environment smoothly

1.5 INTENDED USERS AND USES

The current intended users include the students completing the lab simulation, and possibly the professor to test the lab before it is given to students.

This project is intended to be used as a virtual alternative for students to complete BBMB 102 lab modules.

1.6 ASSUMPTIONS AND LIMITATIONS

Assumptions

- The product will primarily be used by Iowa State students.
- The users have a stable internet connection and reliable computer. (Most is done on the client side)
- The product will not be used on mobile devices.
- The user accesses the website with a [WebGL compatible browser](#).
- User understands the terminology needed to complete the lab module.

Limitations

- The lab procedure must be possible to complete with a mouse and keyboard.
- The product must be small enough in size to fit on our provided virtual server (default 50gb)
- The load time for the product will be less than 5 seconds and the download speed from the server will be better than 3mb/s
- Computer Specs
 - Windows 7 or newer / MacOS Mojave or newer
 - 4 GB RAM

1.7 EXPECTED END PRODUCT AND DELIVERABLES

- Main Website: Possible login for students and professors each with their own actions available. Students should be able to enter any open modules to complete, while professors should be able to either ask for the addition/editing of modules, directly edit some text portions of modules, or open a module for completion for the students.
 - Student Access to the Modules Delivered by End of Fall Semester
 - Following Features may or may not be completed. Should they be finished they will be delivered during the Spring Semester.
- 3~7 Lab Modules: A virtual representation of the real-world equivalent of a BBMB 102 lab. To be seen how many modules can be made and what labs they will be representing. Generally, consists of multiple steps for each lab with an interactive 3D environment to complete the tasks within each step. Includes notes that can be written in or view generated data from the results of their lab.
 - Module 1 should be delivered by the end of Fall Semester
 - Subsequent modules will be later delivered (As seen in section 2.4 Image 2)

2 Project Plan

2.1 TASK DECOMPOSITION

- Page Structure
 - Home page: HTML page loaded from URL, contains links to module pages and about page
 - Module pages: Each module page launches the react script for that module, displays the 3D environment, and allows the user to interact and progress with the experiment.
 - About page: The about page describes the contents and history of the project, contains a link to the GitHub, and contact information for the team.
- 3D Environment
 - Camera controls: The camera controls allow the user to move around the environment by shifting left and right and adjusting the zoom level.
 - Modeled objects: Module loads room and correct objects for the current module.
- Interactive Objects
 - Protective equipment: The user must equip PPE before continuing with the experiment, to teach proper lab safety.
 - Experiment equipment: The lab environment contains models of equipment to be used in the lab module that feature interactivity replicating their real-life counterparts.
 - Script progression on interaction: The script is divided into steps, that progress as the experiment continues, and describe how objects can interact.
 - Interactions between specific objects: Specific objects should execute specific model swapping / data recording values based on their placement.
- Experimental Values & Relations
 - Measurable: Objects should contain metrics for measuring values
 - Accurate: Visible/Measurable values should be accurate to the nature of the experiment.
 - Notebook entry: User should be able to enter data into their notebook and validate whether it is accurate or not.
 - Value graphing: User's notebook should graph input and show trends.
- Notebook
 - Value entry and validation: Users can input the data they collect while completing the module and can see if their results match the lab solutions.
 - Value graphing: Some data input by the user will be displayed in a point graph.
- Help menu
 - List of clear steps: Steps of each module would be displayed in the help menu
 - Completed steps are crossed off: After each step are completed, the step would be crossed automatically
 - List of equipment with pictures: The help menu displays each equipment needed for the experiment with its respective picture
 - Necessary values: The help menu would display any necessary information (e.g. periodic table) needed for the experiment
- User Progress

- Program saved progress by module: The homepage should show whether a signed-in user has completed certain modules.
- Saves user history when a module is completed: The fact that a module has been completed should be safely stored with the user's information.

2.2 RISKS AND RISK MANAGEMENT/MITIGATION

Performance Targets

- Risks:
 - Product will not contain all 7 modules | 0.5
 - Liquid animation will not render properly | 0.8
- Possible Solutions:
 - Minimum number of modules (3)
 - Too be seen with the limitations of 3JS
 - Possible locked view animation taken from elsewhere.

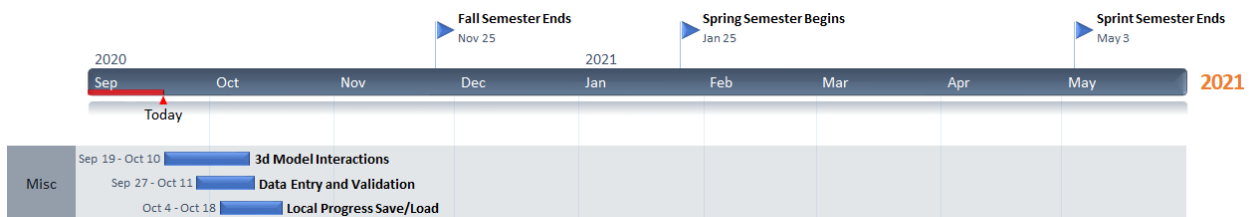
Detailed 3D models

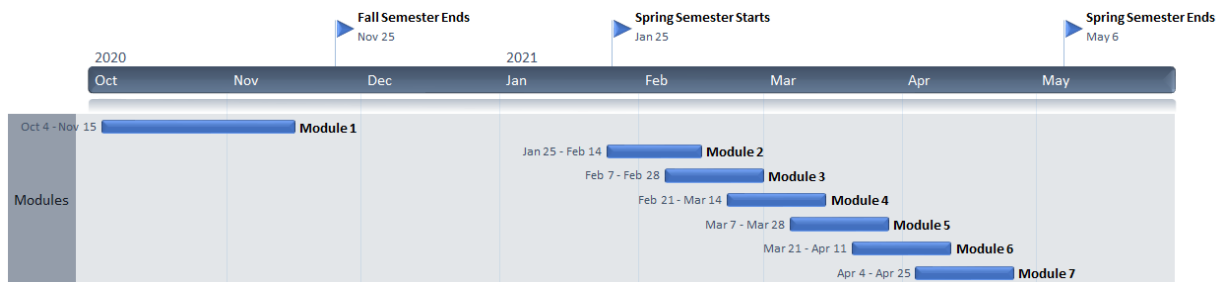
- Risks:
 - File Sizes are too large for running in browser | 0.4
 - Models take too long to load | 0.7
- Possible Solutions:
 - Lower poly count
 - Heavy caching

2.3 PROJECT PROPOSED MILESTONES, METRICS, AND EVALUATION CRITERIA

Some key milestones for the first semester include having a base site up and running, having models created for the first module, accessing the modules, and completing the first module. After these main points are completed the remaining models will be completed and another major milestone for each completed module. Some evaluation criteria within the modules will be various user tasks each experiment within the module and the ability to complete them on the user side.

2.4 PROJECT TIMELINE/SCHEDULE





This timeline may be subject to change as problems or new requirements arise.

2.5 PROJECT TRACKING PROCEDURES

We will primarily use GitLab Boards for project tracking. Tasks will be created in the “TODO” Column, then moved to the “In Progress” Column when a member begins work on the task. Finally, Tasks will be closed when completed.

We will use metrics tracked by GitLab to evaluate our progress.

2.6 PERSONNEL EFFORT REQUIREMENTS

Tasks	Time
3D Model Interaction	Creation of Models: 0-1 person-hours per model depending on available download. Viewing of Models: 0.5 person-hours per model Interacting Models: 1-4 person-hours per model set and number of interactions
Data Entry and Validation	4+ hours per module, all entries must be accurate and precise. Graphs must display information accurately.
User Authenticated Save/Load	10+ person-hours: must set up a backend with authentication, front-end login page, back-and-forth communication, and returning save data.
Modules 1-7	100+ person-hours for the first module (3), and 30+ for each of the rest. 280+ person-hours total.
UI	40+ person-hours. UI must be intuitive and functional. Must be able to load different scenes from the homepage, and open/close the notebook in each module.

2.7 OTHER RESOURCE REQUIREMENTS

We will require server space to allow students access to the virtual lab.

2.8 FINANCIAL REQUIREMENTS

N/A

3 Design

3.1 PREVIOUS WORK AND LITERATURE

A similar project, but for a 2D environment is currently in production by one of our current professor advisors (Prof. Mitra) and his team. We have had a meeting with this team and have been given not only permission to use some of their code, but also advice on how to use it and what may be of use. As this project is a work in progress many of their ideas may be modified or not used at all should they not be compatible with our 3D layout, however also due to their 2D environment their animations and models are easier to create and implement.

As such a special thank you to two of the members, Daniel Han, and Graham T. Mobley, for advising us on their project.

ID	Gel filtration medium	Separation Range (Da)
1	Superdex 75	1,000 - 70,000
2	Superdex 200	25,000 - 699,000

Pick up to 7 standard samples from below: must choose Blue Dextran

ID	Std Sample	Mw (Da)	
1	Vitamin B12	1,350	<input type="checkbox"/>
2	Aprotinin (Bovine Lung)	6,500	<input type="checkbox"/>
3	Myoglobin (Horse)	17,600	<input type="checkbox"/>
4	Carbonic Anhydrase (Bovine Erythrocytes)	29,000	<input type="checkbox"/>
5	Ovalbumin (Chicken Egg)	43,000	<input type="checkbox"/>
6	Hemoglobin (Horse)	65,000	<input type="checkbox"/>
7	Alcohol Dehydrogenase (Yeast)	150,000	<input type="checkbox"/>
8	β -Amylase (Sweet Potato)	200,000	<input type="checkbox"/>
9	Apoferitin (Horse Spleen)	443,000	<input type="checkbox"/>
10	Thyroglobulin (Bovine Thyroid)	669,000	<input type="checkbox"/>
11	Blue Dextran 2000	2,000,000	<input type="checkbox"/>

<https://mitraresearch03.cs.iastate.edu/modules/module3>

3.2 DESIGN THINKING

A lot of our design thinking regarding the 3d environment is driven by the requirement to keep the application light in order to run in browsers on less powerful devices.

The overall driver for the lab design is guiding the user to complete the lab. It should be intuitive for the user to understand what actions they need to take next, and it should not be possible for the user to take actions that would result in not being able to advance.

3.3 PROPOSED DESIGN

The current design implements 3JS, React, and other items to create a webpage with an HTML canvas. The canvas is what the modules use to display the 3D lab environment to the user and allow them to interact with objects in order to complete various tasks for the lab. The viewing of the 3D models, and the catching of their “interactions” is completed by checking if the boundary boxes of their models overlap. This approach was only slightly tested but has been implemented enough for a client demonstration and with their approval continued for the rest of the project.

We are also implementing ESLint, an in-program static code analysis tool to help us identify problems within our JavaScript. This is mostly used to make sure our team upholds specific coding standards such as JSDoc (commenting code), indentation, and other styling issues. The standard we applied is Google’s JavaScript style guide.

3.4 TECHNOLOGY CONSIDERATIONS

Strengths:

- Allows students to complete labs in an interactive environment.
- Labs can be completed and repeated much quicker than a normal lab would take. (Most of the waiting for a reaction to happen has been removed)
- Allows for safe learning during the epidemic currently happening (COVID) or on normal times allowing a sick student to participate in a lab they may have missed.

Weaknesses:

- The labs are individual as the 3D environment is client side.
 - Have the students use some other screen share software to complete the lab together. Maybe include some way for the person actually running the lab to state the partner who helped them or give them a code to allow the partners to view the updating data on their own screens.
- There is less hands-on experience for the students.
 - This is meant to be an alternative, there is no solution for hands-on experience other than making them go to an actual lab after this one with the experience gained from this software.
- The virtual lab runs in a web browser.
 - This limits the capabilities of the software by restricting some hardware use, such as the full graphics card capabilities. This could be mitigated by creating the software as an application but would limit the program’s accessibility, since multiple versions of the software would need to be created for different operating systems.

3.5 DESIGN ANALYSIS

At the moment the moveable object with multiple textures do not move all parts at the same time. We believe this is caused since different textures are create multiple meshes. This problem is likely to be resolved but has not as of this moment.

Otherwise most of the knowledge needed for various items has been found but not always implemented yet. Ex. There is object collision detection, but items can pass through one another if they don’t have an action associated with them.

As our development process is meant to be agile, there will be many small iterations with improvements to the design. Should an item pose problems or create errors, an alternative idea will be created at that time.

3.6 DEVELOPMENT PROCESS

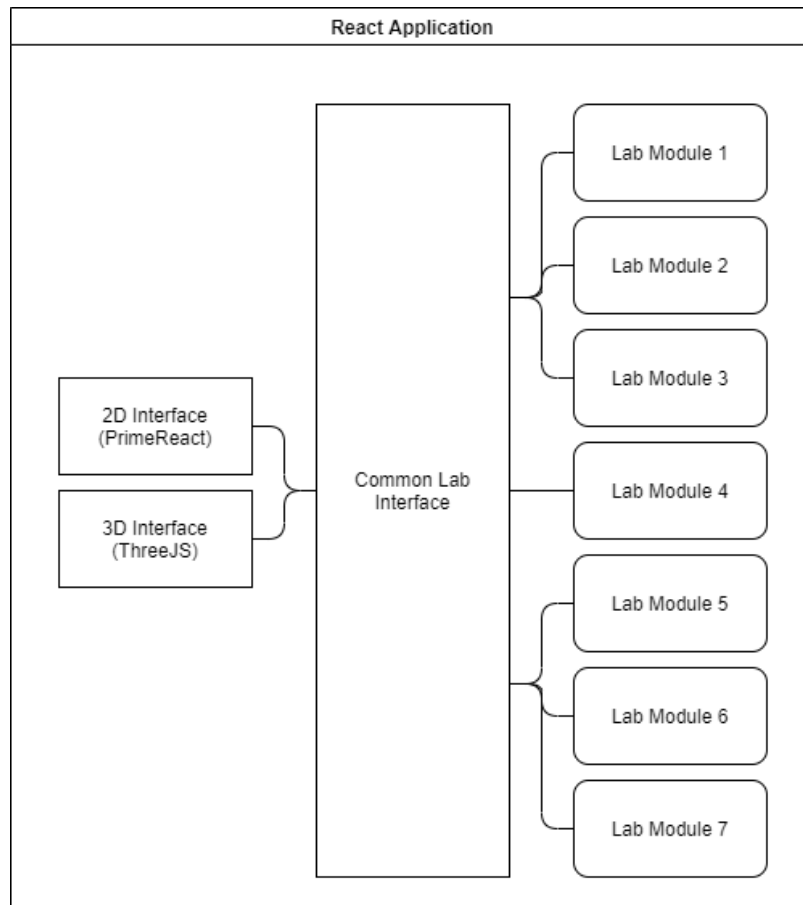
Our development process is based on the Agile method. We hold two weekly standup meetings where we go over what was completed since the last meeting, ask/answer questions to our client, and assign new tasks. As this project is entirely software based, we believe this method will allow us to produce the best results while listening to our client's needs.

3.7 DESIGN PLAN

Our plan is to create something like a game engine and implement the various modules as "games". Each module will be independent of the other modules. The "engine" will handle tasks like loading 3d models, detecting changes the user has made, and updating various components. The modules will primarily consist of callbacks for actions the user has taken.

There are two areas where the module will have the freedom to define what exists and where; the 3D environment and the notebook (a 2d overlay the user can toggle visibility of). Other than these two, the various other elements will be controlled by the main application.

We chose this combination of PrimeReact and ThreeJS because they met the needs of our assumptions and limitations. While many combinations of dependencies and languages would have met our needs, we chose to use ones that our members had previous experience with to reduce the amount of additional learning needed.



4 Testing

4.1 UNIT TESTING

- UI testing
 - Interface must advance correctly when user completes a step/module
 - Ensure user cannot get stuck in any step - can always advance or reset if needed
- Data Validation
 - All data must be correctly stored and validated
 - Step can only be completed when correct data is entered
 - Ensure max/min are set, only proper data format is entered
- Object Data
 - Collisions must be consistently detected
 - Positions must be updated immediately

4.2 INTERFACE TESTING

Front-End and Back-End Connectivity: In order to test the interface between the two sides, tests will be conducted using Postman. Tests will be written to validate server execution, error handling, and connection resets and failure scenarios.

4.3 ACCEPTANCE TESTING

We hold demonstrations of our current progress every two weeks to ensure the project is advancing in a manner acceptable to the client. We also meet with our client weekly to ensure our specifications are as up to date as possible, and any deviations from the original design are approved.

4.4 RESULTS

Manual Testing

- Ability to drag object
 - Successes:
 - Singular objects can be moved within the 3D environment
 - Objects can be locked in place if desired
 - Failures:
 - Objects made of multiple meshes (due to parts or multiple materials) do not move as one.
- Object collision
 - Successes:
 - Bounding boxes around objects correctly recognize and log when another box intersects with it.
 - Failures:
 - Objects can phase through each other and the lab environment as limits and physical collision have not been implemented.

Automatic testing

- ESLint Styling
 - Successes:
 - Is able to catch styling errors in code and will force us to conform to Google's JavaScript style guide. This includes correctly commenting code.
 - Failures:
 - None so far.

5 Implementation

So far, for the 3D animation, we have implemented a virtual work lab station that includes two lab benches, a countertop sink for draining liquids at the back of the room and flasks. For the next semester, we plan on starting to implement script for each module starting with module 3 as our client requested. Once module 3 script is completed, the implementation for the rest of the modules will be easier to follow as most of the script looks similar.

For the 2D part of the project, currently we can iterate through actions and steps of each module. For next semester we plan on implementing more script like dialog selection, chart, and graph.

6 Closing Material

6.1 CONCLUSION

As of the nearing end for this semester the group we have completed a three-dimensional rendering of a lab station, and access to it through lab modules. In this virtual environment we have been able to create multiple lab equipment models and have the product notice when they are close enough to interact with each other. While the original plan for starting module one before the semester has ended has yet to be reached, we believe we have created a strong foundation to base the lab modules upon.

6.2 REFERENCES

Mrdoob et al. (2020) three.js Library (v0.120.1) [Library] threejs.org

Rickhanlonii et al. (2020) React Library (v17.0.1) [Library] reactjs.org

Primtetek (2020) PrimeReact (v5.0.1) [Library] primefaces.org

6.3 APPENDICES

Our software will be hosted on an ISU VM at sdmay21-12.ece.iastate.edu

3D virtual lab screenshots: The left end contains the 3D animation, and the right end contains the 2D part of the project.

