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: 10/3/2020

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 9

2.8 Financial Requirements 9

3 Design 9

3.1 Previous Work And Literature 9

3.2 Design Thinking 9

3.3 Proposed Design 9

3.4 Technology Considerations 9

3.5 Design Analysis 10

3.6 Development Process 10

3.7 Design Plan 10

4 Testing 10

4.1 Unit Testing 11

4.2 Interface Testing 11

4.3 Acceptance Testing 11

4.4 Results 11

5 Implementation 11

6 Closing Material 12

6.1 Conclusion 12

6.2 References 12

6.3 Appendices 12

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

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

List all requirements for your project – functional requirements within your project context, economic/market requirements, environmental requirements, UI requirements, and any others relevant to your project.

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.

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 and 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.

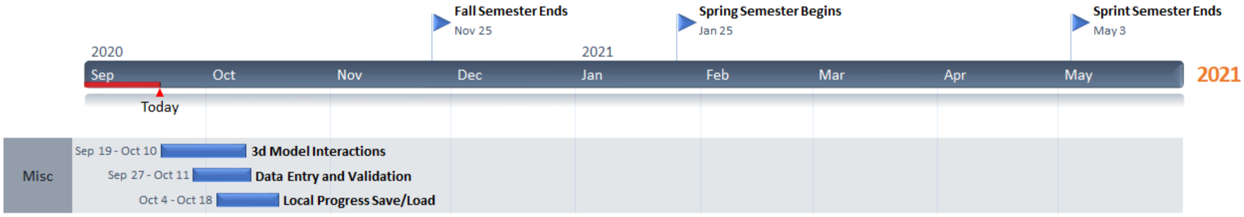
Detailed 3D models

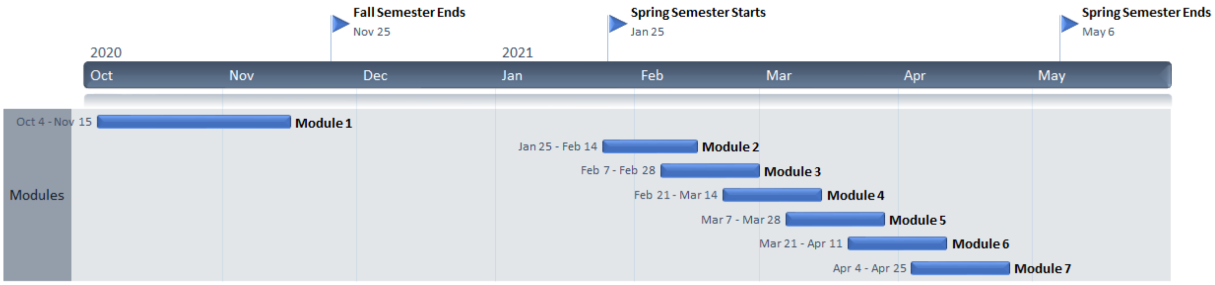
- Risks:
    - File Sizes are too large for running in browser | 0.4
    - Models take too long to load | 0.7
- Possible Solutions:
    - Lower poly count
    - Heavy caching

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.

2.6 Personnel Effort Requirements

Include a detailed estimate in the form of a table accompanied by a textual reference and explanation. This estimate shall be done on a task-by-task basis and should be the projected effort in total number of person-hours required to perform the task.

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

Include relevant background/literature review for the project

– If similar products exist in the market, describe what has already been done

– If you are following previous work, cite that and discuss the **advantages/shortcomings**

– Note that while you are not expected to "compete" with other existing products / research groups, you should be able to differentiate your project from what is available

Detail any similar products or research done on this topic previously. Please cite your sources and include them in your references. All figures must be captioned and referenced in your text.

3.2 Design Thinking

Detail any design thinking driven design "define" aspects that shape your design. Enumerate some of the other design choices that came up in your design thinking "ideate" phase.

3.3 Proposed Design

Include any/all possible methods of approach to solving the problem:

- Discuss what you have done so far – what have you tried/implemented/tested?
- Some discussion of how this design satisfies the **functional and non-functional requirements** of the project.
- If any **standards** are relevant to your project (e.g. IEEE standards, NIST standards) discuss the applicability of those standards here
- This design description should be in **sufficient detail** that another team of engineers can look through it and implement it.

3.4 Technology Considerations

Highlight the strengths, weakness, and trade-offs made in technology available.

Discuss possible solutions and design alternatives

3.5 Design Analysis

- Did your proposed design from 3.3 work? Why or why not?
- What are your observations, thoughts, and ideas to modify or iterate over the design?

3.6 Development Process

Discuss what development process you are following with a rationale for it – Waterfall, TDD, Agile. Note that this is not necessarily only for software projects. Development processes are applicable for all design projects.

3.7 Design Plan

Describe a design plan with respect to use-cases within the context of requirements, modules in your design (dependency/concurrency of modules through a module diagram, interfaces, architectural overview), module constraints tied to requirements.

4 Testing

Testing is an **extremely** important component of most projects, whether it involves a circuit, a process, or software.

1. Define the needed types of tests (unit testing for modules, integrity testing for interfaces, user-study or acceptance testing for functional and non-functional requirements).
2. Define/identify the individual items/units and interfaces to be tested.
3. Define, design, and develop the actual test cases.
4. Determine the anticipated test results for each test case
5. Perform the actual tests.
6. Evaluate the actual test results.
7. Make the necessary changes to the product being tested

8. Perform any necessary retesting
9. Document the entire testing process and its results

Include Functional and Non-Functional Testing, Modeling and Simulations, challenges you have determined.

4.1 Unit Testing

– Discuss any hardware/software units being tested in isolation

4.2 Interface Testing

– Discuss how the composition of two or more units (interfaces) are to be tested. Enumerate all the relevant interfaces in your design.

4.3 Acceptance Testing

How will you demonstrate that the design requirements, both functional and non-functional are being met? How would you involve your client in the acceptance testing?

4.4 Results

– List and explain any and all results obtained so far during the testing phase

- Include failures and successes
- Explain what you learned and how you are planning to change the design iteratively as you progress with your project
- If you are including figures, please include captions and cite it in the text

5 Implementation

Describe any (preliminary) implementation plan for the next semester for your proposed design in 3.3.

6 Closing Material

6.1 Conclusion

Summarize the work you have done so far. Briefly re-iterate your goals. Then, re-iterate the best plan of action (or solution) to achieving your goals and indicate why this surpasses all other possible solutions tested.

6.2 References

List technical references and related work / market survey references. Do professional citation style (ex. IEEE).

6.3 Appendices

Any additional information that would be helpful to the evaluation of your design document.

If you have any large graphs, tables, or similar data that does not directly pertain to the problem but helps support it, include it here. This would also be a good area to include hardware/software manuals used. May include CAD files, circuit schematics, layout etc,. PCB testing issues etc., Software bugs etc.