Software Design Document

February 16, 2024

Version 2.0

Sponsored by: Dr. Elise Donovan, Sneha Vissa, Adonna Rometo

Mentor: Tayyaba Shaheen

Team members: Dayra Quinonez, Nicole Sylvester,
Rino De Guzman, Bailey Rosato

Table of Contents

1. Introduction

The BIO201, Anatomy and Physiology course at Northern Arizona University (NAU) stands as a fundamental requirement for aspiring medical students as it provides in-depth education of human anatomy and physiology. However, there is currently a noticeable lack of diversity in the educational content that students are studying, impacting the overall quality of patient care and limiting students to generic, predominantly white, and slender anatomical models.

Our clients, Dr. Elise Donovan, Adonna Rometo, and Sneha Vissa, professors in the Department of Biology Sciences and members of the Biological Sciences Inclusive Excellence (IE) Committee, are also the coordinators of the anatomy and physiology courses at NAU. Our clients have evaluated their courses and identified a lack of diversity in their current anatomical models. Currently, the sole non-white model at NAU is a torso, and the scarcity and high costs of non-white models pose significant challenges for our clients. Consequently, our clients are left with limited teaching materials, including a poorly constructed partial model or expensive virtual alternatives. Our clients want students taking anatomy and physiology to feel represented and have a stronger connection to course material by fostering inclusivity with more diverse models.

Our solution to address our clients needs is to develop a supplemental web application lab resource that will showcase anatomically accurate, three-dimensional models that users can customize. Users can change the biological gender, body size, and skin tones of the models facilitating more diverse representations. These diverse models will serve as a valuable study tool for students during lectures, labs, and SI sessions. The application will tailor models and unit information based on material being taught specifically in NAU's BIO201 course.

The requirements of our web application have evolved through collaboration with our clients at weekly meetings. Key user-level requirements include a user-friendly, interactive interface, student centric exploration that empowers students to explore diverse models, dynamic and interactive customization that allows users to adjust model parameters, and lastly, information accuracy and alignment with the current BIO201 curriculum. In order to accomplish these goals, our team has also decided on key functional and performance requirements. The key functional requirements focus on a descriptive content menu, content display and interaction, the ability to export models, and a user guide to help students navigate the application. In terms of performance, the most significant concerns are content and model accuracy and usability to

ensure that the models are beneficial to students' learning. The three ways we plan to prevent environmental constraints are to use open source technologies with strong use communities, maintain a well-documented codebase that tracks technology versions, and create an extensible application that contains a modular code base for future changes. To summarize, the web application solution that addresses our clients' need for diversified anatomy and physiology models includes:

- A user-friendly, highly interactive interface
    - Descriptive content menu
    - A user guide
- Student centric exploration that allows students to explore more diverse models
    - Ability to export models
- Dynamic customization that allows users to change model parameters including:
    - Skin tone
    - Biological gender
    - Body type
- Information accuracy and alignment with current BIO201 content
    - Content display

2. Implementation Overview

To give an overview of the solution that the team envisioned, the product's goal is to create a supplemental web application that accompanies the BIO 201 course curriculum. This web application will allow students to generate models with customizable features such as skin tone, body size, and biological gender. Blender will be used to create high-resolution 3D models for the web application. The created models are then stored in the database and displayed in the front-end through the use of Three.js. To store all the units, subunits, educational content, and the models themselves, Firebase will be used as the product's database. Lastly, the application will be using React with a combination of Google's MaterialUI to create a fluid and interactive front-end for the user.

Through precarious research during the Fall 2023 semester, the technologies that align best with the solution vision of the team and the clients include Firebase, Blender, Three.js, and React with MaterialUI.

Blender

In order to provide users with reliable and accurate learning material, the 3D models must be realistic and high-resolution. It is also necessary that the chosen 3D model renderer program is able to export its renders via .gltf files, this allows the rendered files to be compatible with other webGL rendering applications. For that purpose, the team is using Blender, a free and open-source 3D creation suite.

Firebase

To ensure that the product will contain all the necessary educational content and data for the models, choosing the right database is essential. The database technology must be scalable as this product will be expanded upon in the future and it must be able to handle the growth in data without sacrificing performance. Firebase's Realtime data synchronization is perfect for ensuring the user's interaction with the application is as responsive as possible. The serverless structure of Firebase will also reduce operational overload

React

React is a Javascript framework that offers incredible user interaction support and scalability since it excels in responsive design due to its virtual DOM, component

architecture, and interactivity tools. This framework allows for a seamless and easy-to-use front-end for the user.

Three.js

Three.js was created by Ricardo Cabella and uses WebGL to draw 3D in most cases. Three.js is an incredibly popular library that will allow easy import of Blender .gltf model files to display onto the web application. This will allow users to interact with a 3D model such as changing body size, biological sex, and skin tone.

3. Architectural Overview

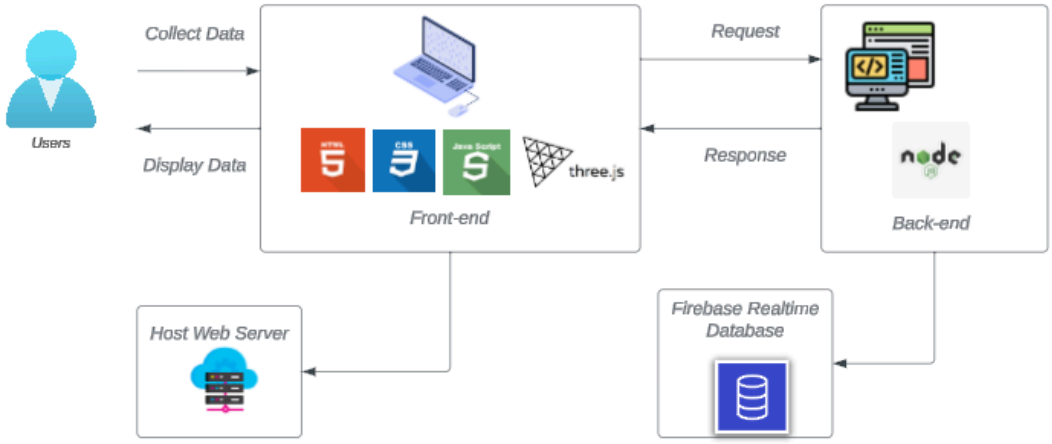

Figure 3.0

The first module of the architecture diagram in Figure 3.0 is the front-end component, which is the user interface of the application. Front-end is responsible for presenting information that is available to the user and what they are able to interact with. As seen in the diagram above, the front-end is connected to a server, this server allows users to enter the web application through a linked URL. These pages are set up through a combination of high-level languages: JavaScript, HTML, and CSS. Within JavaScript, the React library is being utilized. The web page that the user first encounters is the course entry page. The course entry page is set up to display a typical log-in page that requires users to input and then submit a course code to access the rest of the web application. Upon successful input, the next page the user is redirected to is the main menu that displays the course content units and subunits the users are learning in their BIO201 course. The users are able to interact with the page by clicking on the drop-down menus for each unit and a button for the user guide. When the user clicks on a certain subunit, the user will be redirected to the Subunit's page. Within the Subunit page, course material as well as a three-dimensional anatomical model are rendered. Model customization UI elements displayed

will allow the users to modify and create their own models. These elements include a color picker, drop-down selector, and toggle switch to be able to change the color tone, body size, and biological sex of the model respectively. Users will also be displayed a button to export their models and reset the model view.

As the user is interacting with the application, the front-end fetches data from the backend to dynamically allow functionality and display content. The back-end responds to requests, allowing the user to have a smooth experience. It is responsible for managing and processing unit and subunit data including all of the functionality that the users are interacting with from the course entry page to the Subunit pages. For example, the three-dimensional models are stored in the backend through the utilization of Node.js. Three.js displays the correct model files on the front-end Subunit page and the back-end retrieves the correct model files for the specific unit. The backend retrieves content to display to the users dynamically, such as the user guide pdf, .gltf model files, and course content.

Transitioning from dynamic data retrieval to data storage, all data is stored within the database, which is depicted as the final section in the application architecture diagram. A database is a system that stores information for access, update, or analysis. As indicated, the database is essential as it holds the necessary information for the project. The Firebase Realtime Database, as shown in the diagram, will manage this section. The database, set up as a JSON file, will hold different information on each page. The main menu will access the database for unit and subunit titles. The Subunit pages will retrieve model files, subunit content descriptions, and images corresponding to the course material. These pages also feature a dropdown menu, requiring the units and subunit titles to be retrieved as well. The system utilized a Client-Server architecture style. The Client-Server architecture is a model where tasks are distributed through servers and clients. The servers are the providers and the clients are the requesters. The front-end is acting as the client while the back-end, including the database, is the component that is acting as the server.

4. Module and Interface Descriptions

The architecture of the application can be further divided into specific components of the interface and back-end functionality. The following section will be organized into the flow of use for the web application, starting with the login page.

4.1 Course Entry

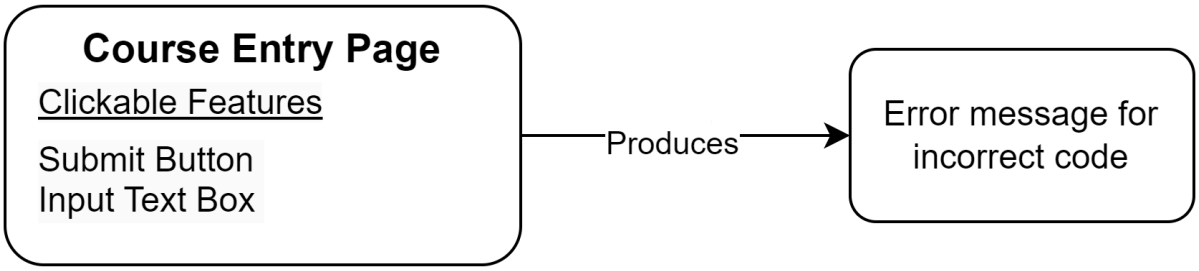

Figure 4.0

The responsibilities of the course login page are to allow users to enter a course code to access the features of the website. Once a valid code is entered, users are able to proceed to the main menu of the application and navigate to any page. If an incorrect code is entered, a window alert will pop up, informing the user that the code was incorrect. The user will be allowed to re-enter a code after the alert to gain access. The physical interface of this component is a page that has a text box and a submission button for the course code. This component works to ensure that the application is used by NAU students/faculty to limit the amount of users. There is also an image specific to BIO201 that is displayed alongside the login text input.

4.2 Main Menu

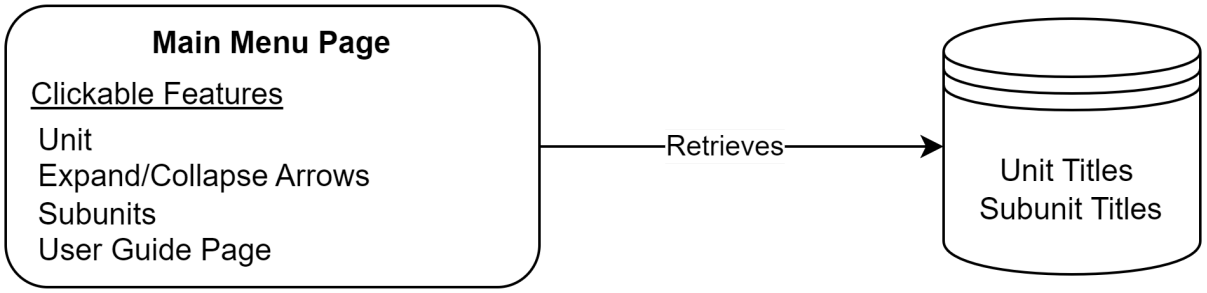

Figure 4.1

The responsibilities of the main menu page are to list the units and subunits of the BIO201 curriculum, as well as the user guide. The units and subunits correspond to the BIO201 lab manual and have been approved by our clients. The retrieval of units and subunits for the main menu is done with the help of the database. The main menu references the courseData key in the database and returns the unit and subunit titles that are then displayed to the user in a menu form. The main units are expandable to show their corresponding subunits. The user guide is also a page that the users can access from the main menu, which includes downloading a PDF of the user guide. When a user selects a subunit, they are navigated to a Subunit page that displays information specific to that subunit alongside a 3D model.

4.3 User Guide

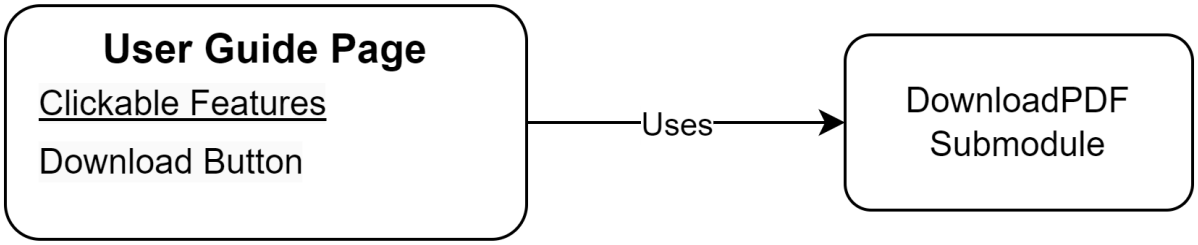

Figure 4.2

The user guide page is responsible for providing easy access to the user guide PDF in the form of a download button that users will be able to locate if they are in need of assistance. The PDF will provide information in text as well as images on how to navigate through the application. It will be available for users to resolve questions about the application and enhance the user experience. The user guide page contains the DownloadPDF module, which is responsible for saving a PDF of the user guide on the user's device.

4.4 Subunit

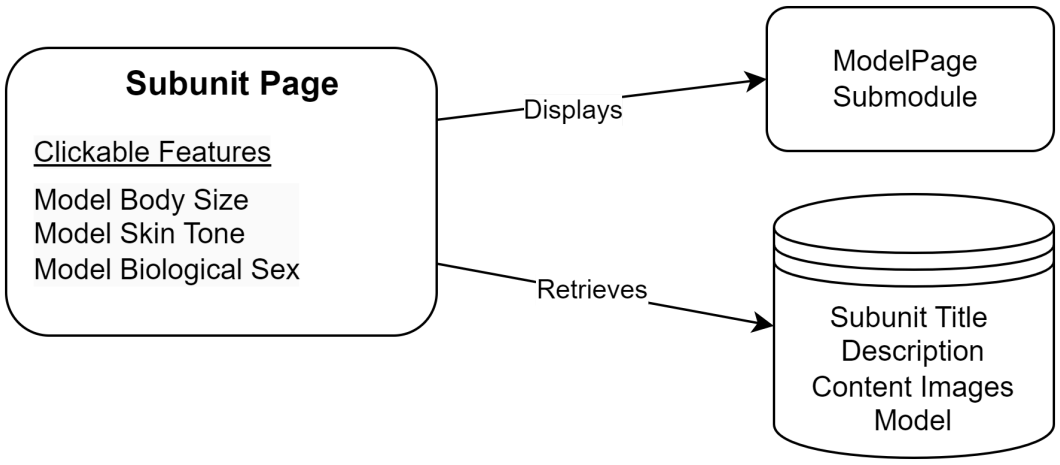

Figure 4.3

The Subunit page will contain all the relevant course content relating to the specific unit as well as the 3D model for the users to interact with. The services provided by this module are an interactive interface with the ability to modify specific components of the model, such as body size, skin tone, and biological sex in conjunction with the course content. The Subunit page displays the ModelPage submodule, which is part of the larger ModelRendering module.

4.5 Sidebar Menu

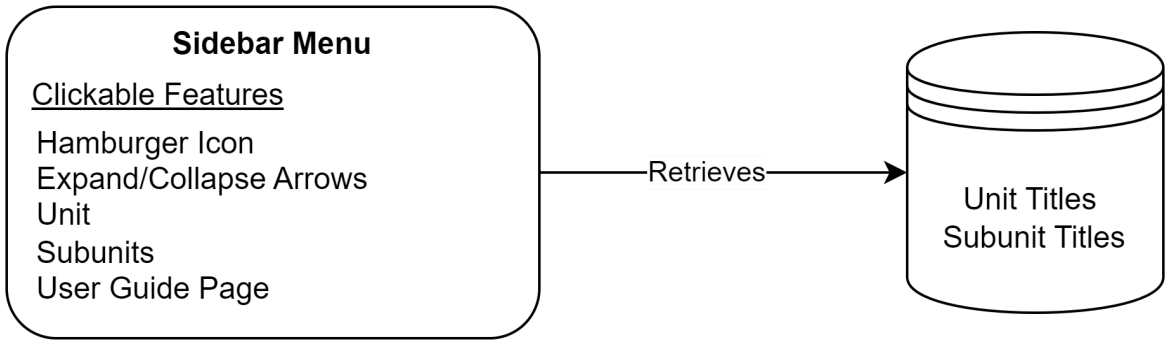

Figure 4.4

Included within the Subunit component, there is also a pop-up sidebar menu that is responsible for redirecting users to the desired subunit. It is a condensed version of the main menu that allows users to switch subunits from the Subunit page, rather than navigating back to

the main menu. Like the main menu, the sidebar menu references the courseData key in the database and returns the unit and subunit titles that are then displayed to the user in a menu form.

4.6 Model Rendering

The responsibilities of the Model Rendering component are to render the 3D model for users when the Subunit page is first opened and update when the users make changes to the model parameters. It also allows the user to export the model as a PNG. Within the Model Rendering component, there are submodules that all work together to ensure a seamless display of the model on different-sized screens. The physical interface of the Model Rendering component is a rendered model of a human body that can be zoomed in/out on and rotated. Users can view the model to apply the course content that is displayed on the same page.

The ModelRendering module is broken down into submodules:

AnatomyModel

AnatomyModel is responsible for rendering the 3D models that represent female and male models, with different body sizes. The model is loaded using a glTF file loader and the model's appearance can be customized based on values from the CharacterCustomizationContext module. Functions from the CharacterCustomizationContext hook are used to set the skin tone, body size, and gender by extracting materials of the model.

CameraControls

CameraControls is responsible for OrbitControls of a Three.js scene. These enable the rotate, zoom, and panning features for users to manipulate the model view within the scene. On the physical interface, it allows users to scroll, zoom in and out, right-click and drag to pan, and left-click and drag to rotate the models.

CharacterCustomizationContext

CharacterCustomizationContext is responsible for managing various state variables related to model customization. These include variables that set the mode of the camera, the model's skin tone, the biological gender of the model, and the model's body size. This file also specifies a range of 14 hex colors to be used in the skin tone color selector as preset color options.

Experience

The Experience submodule is responsible for creating the scene for the 3D model. This scene acts as the rendering space of the model. It includes the CameraControls component that provides interactive controls allowing the user to manipulate the view of the model. The lights for the scene are also set, which illuminates the model within the scene. It has two directional lights that light the model from the front and back. The Experience module also includes the AnatomyModel module that renders the 3D model within the scene. The Experience module is then exported to use in the ModelPage so that the scene can be visible and interactive to users from the Subunit page.

SettingsConfigurator

The SettingsConfigurator submodule is used within the Interface submodule and represents the setting panel for model customization. It includes UI elements such as a skin tone selector element, a switch element to select the biological sex of the model, and an element for selecting body size. This component also imports female and male icons to use within the biological sex selector.

Interface

The Interface submodule is responsible for displaying the UI elements that allow the user to customize the model. The Interface component is a functional component and a top layer container is used to structure the UI layout. First, A button component is rendered that, when clicked, toggles the components in the SettingsConfigurator that renders UI elements for all the model parameters that can be changed.

ModelPage

The ModelPage submodule combines a 3D scene, handled by the Experience module, and a user interface, handled by the Interface component, to create a single structure that allows for the integration of a 3D model with interactive controls into the Subunit Page.

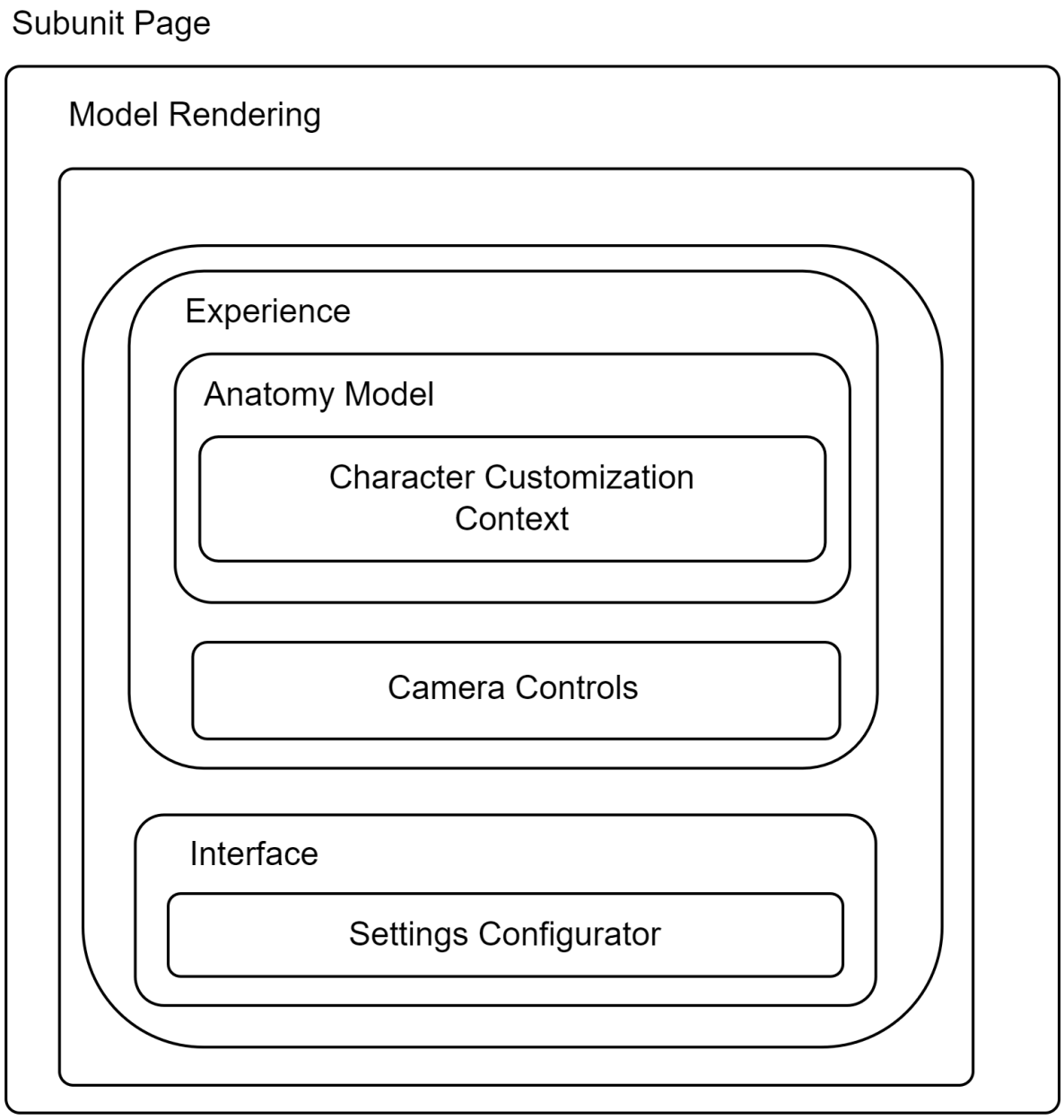

Figure 4.5

All of the above submodules work together to ensure proper model rendering and intractability. As seen in Figure 4.5 above, these submodules work together to provide a seamless user experience.

4.7 Firebase Realtime Database

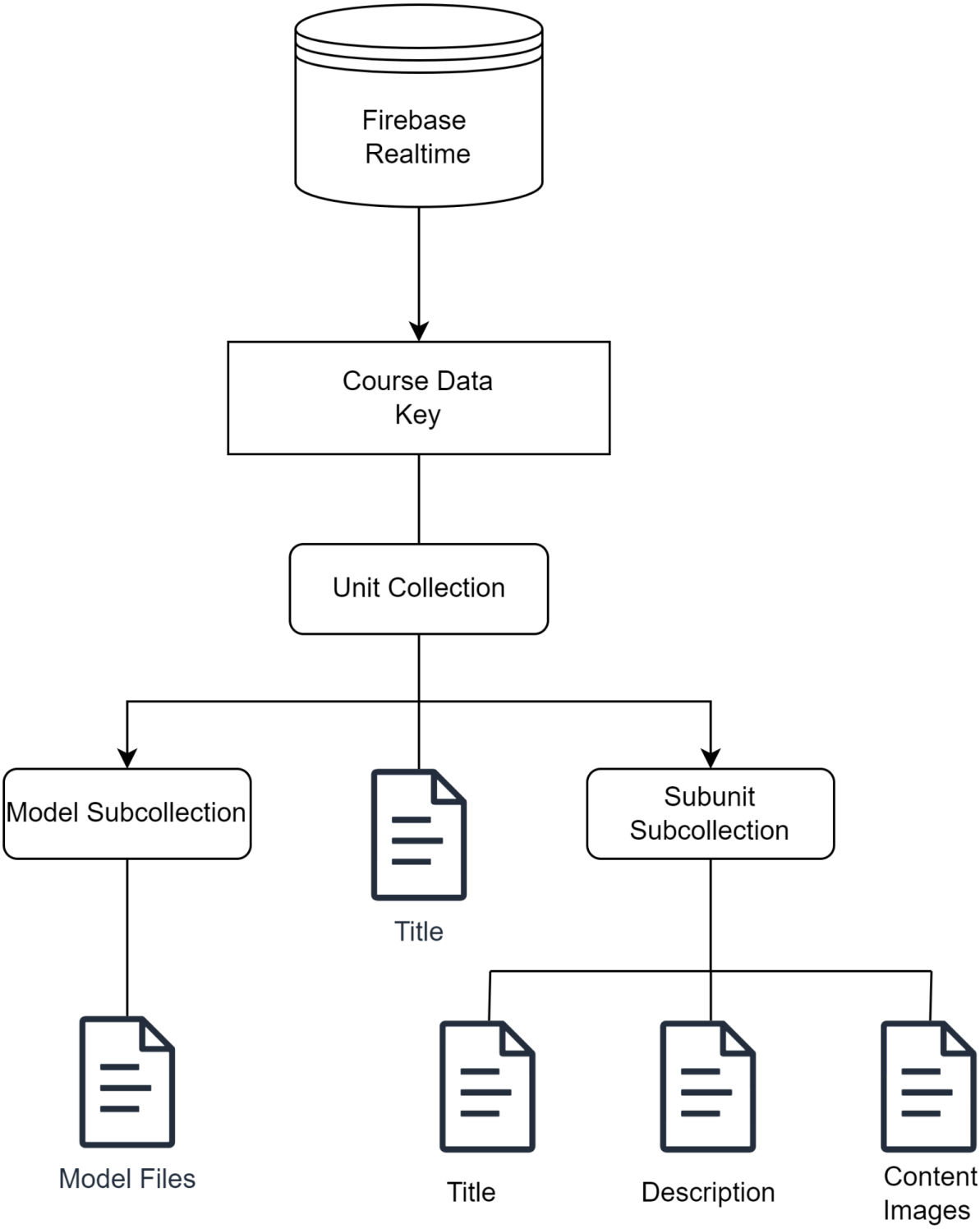

Figure 4.6

The database is organized in JSON files based on the category of data. Data retrieved from the database will be split into a courseData key that is divided into units. Each unit collection can be broken down into its respective subunits and unit title. The subunit subcollection includes its title, description, and in some cases images that are part of the course content. Additionally, for each unit there is a model subcollection that stores the model files specific to a unit of course content.

As demonstrated in Figure 4.6, as a non-relational database, the documents for a category of data will be available once a specific subcollection is accessed. The unit and subunit titles and course content will be organized into plain-text JSON files. The model data is organized into an array, where the columns and rows are representative of ethnicity and body size. Using the array data structure simplifies the retrieval of the specific model types to facilitate faster load times.

4.8 Summary

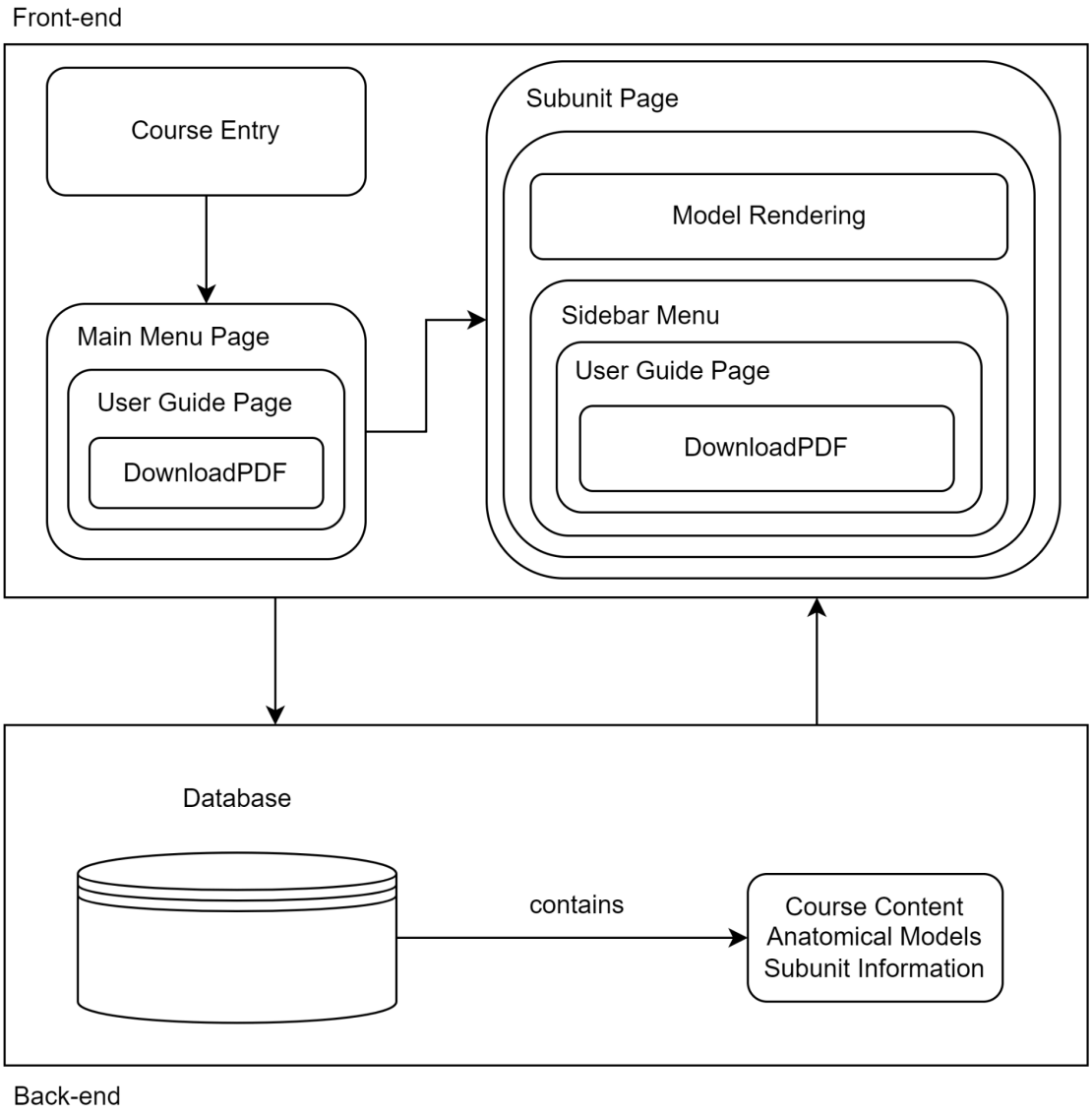

Figure 4.7

To summarize the responsibilities of the front-end modules, Figure 4.7 depicts the interactions between the Course Entry, Main Menu, and Subunit page along with their submodules. To simplify the functions of the Model Rendering submodules and improve the readability of the diagram, they have been abstracted into the Model Rendering module. The Subunit page also includes the Sidebar Menu, which can be used to access other units and the user guide, similar to the Main Menu. To retrieve data about the units for the main menu, the front-end must make a request to the database. This also applies to the model data and course content.

5. Implementation Plan

As there are many subcomponents within the architecture of the application, it allows for development to be conducted in parallel. Currently, there are bi-weekly sprints in which team members will be assigned tasks depending on the progress of the application. Once a week during team meetings, an update will be given from each team member regarding their status and/or struggles that they may have with their tasks. If a task is completed, it will be held for review by at least one other team member. On the other hand, if a sprint is not able to be completed on time, the following sprint will be appropriately adjusted to ensure that the team stays on track with development.

Figure 5.0

The Gantt chart in Figure 5.0 depicts the sprints that the team is currently working on and will be separated into 4 main sections: the model rendering/creation, the web application's front-end, back-end, and unit testing. As you can see from Figure 5.0, the model rendering and creation are being developed in parallel with the application's database from week 3 to week 8. The next phase of development includes the User Guide Content and Unit testing, the development of these functionalities will be conducted parallel as well.

6. Conclusion

In conclusion, our project's aim to address the diversity gap in anatomy and physiology education at NAU has led us to envision and design a web application. This solution, centered around a user-friendly interface, is geared towards empowering students by providing them with anatomically accurate, three-dimensional models that can be customized to reflect diverse characteristics such as biological gender, body size, and skin tones. Our architectural approach leverages a combination of technologies, with Firebase as our backend for data storage and retrieval, React ensuring a responsive design, Blender facilitating the creation of three-dimensional anatomical models, and Three.js integrating these models for display in the web application. The architectural overview showcases a defined flow of pages, starting from the course entry page, through a main menu, to detailed subunit pages where students can interact with the three-dimensional models. The communication between the front-end and back-end is designed to ensure a smooth user experience. Detailed module descriptions further specify the functionality of each component, ranging from the course entry page for access control to the subunit pages where students engage with course content and models. Our implementation plan outlines a design centric timeline, emphasizing parallel development of various components. Team members will collaborate to work on database development, front-end development, course entry, Subunit pages, a main menu, navigation, and a user guide. In essence, this project will address immediate needs identified by our clients and also foster inclusivity and representation in anatomy and physiology.