ÇANKAYA UNIVERSITY

FACULTY OF ENGINEERING

COMPUTER ENGINEERING DEPARTMENT

CENG 407 - System Architecture & Design Decisions

VR Anatomy: VR Based Educational Interactive Human Anatomy Training Platform

Members:

Kutay Ayoğlu – 202111019

Çağla Pelin Doğan – 202111026

Elifnaz Talas – 202111069

Öykü Çoban – 202111066

Sena Akbaba – 202228003

Table of Contents

1.Overview

1.1 Purpose

The purpose of this report is to document the system architecture and the key design decisions of VR Anatomy. It explains how the Unity VR client, the packaged learning content, and the AI Tutor backend work together. It also states the reasons behind major choices such as scene organization, content storage format, and the separation between learning modes.

1.2 System Summary

VR Anatomy is a single user VR application designed for classroom guided use with one headset. The student is the primary user who interacts inside VR. The instructor is a secondary user who guides verbally and does not interact with the application directly.

Core features focus on module based anatomy exploration and a quiz mode that can be started from the main menu.

Main user capabilities are listed below.
• Select a module from the VR main menu
• Explore a 3D anatomy model in VR
• Select an anatomical structure and see it highlighted
• Perform basic interactions such as grab, rotate, and release
• View labels and short info cards for selected structures
• Start Quiz Mode from the main menu independent from modules
• Start Review Mode where only selected model assets are shown and info cards are not shown
• The AI Tutor provides a dedicated chat and voice interface that is opened from the main menu via the “AI Tutor” button.
• The student can ask questions either by typing or by using push-to-talk (speech-to-text).
• The backend returns a short text answer grounded in the project’s learning content using a Retrieval-Augmented Generation (RAG) approach.
• The answer can optionally be read aloud via a speaker button (text-to-speech).
• In this version, the AI Tutor is not structure-aware and does not automatically use the currently selected anatomical structure as context.
• If the AI service is unavailable, the AI Tutor feature is disabled while core VR exploration and quiz features continue.

1.3 Architecture Goals

The architecture was designed to support an educational VR experience that is stable in classroom use and easy to expand with new modules and content. The system includes an AI Tutor feature that may become unavailable due to network or service issues. For this reason, the architecture separates the VR learning functions from the AI Tutor integration.

The main goals are listed below.

• Keep interactive exploration, Quiz Mode, and Review Mode clearly separated so that each mode remains simple and consistent

• Keep core VR functions usable when the AI Tutor service is unavailable, such as module selection, 3D exploration, structure highlighting, and basic interactions

• Allow content updates by changing JSON files and Unity model assets without changing the application logic

• Keep runtime behavior predictable for a single headset and a single student session

• Make the flow of information clear between user interface, content loading, interaction handling, and AI Tutor requests to simplify testing and troubleshooting

• Reduce reliance on online services so that non AI functions can still run in offline classroom conditions

2. High Level Architecture

This section describes the main building blocks of VR Anatomy and how data moves between them during key user actions. The system is centered on a Unity based VR client that loads local content for learning and calls an Azure based AI backend for the AI Tutor feature when available.

2.1 Architecture Diagram and Explanation

The diagram below shows the high-level components and their connections. The VR client runs on the headset, loads JSON and 3D assets locally, and communicates with the AI backend only for AI Tutor chat requests.

Figure 1. High-level system architecture of the VR Anatomy.

The architecture follows a simple rule. The VR learning functions rely on local assets so they can work without network access. The AI Tutor feature relies on the backend and may become unavailable while the VR client continues.

2.2 Main Components

Unity VR Client:
The VR client is the primary runtime environment. It renders the 3D anatomy scenes, handles XR input (controllers/hands), and displays all in-app UI. The client provides the main user modes: Start Learning, Free Review, Quiz, Settings/About, and a dedicated AI Tutor screen. The VR client is designed to function offline for core exploration and quiz features.

Main Menu and Global Screens:
The main menu is the entry point for navigation across the application. From this screen, the student can select modules for learning/review, start the quiz mode, open settings and the about page, or access the AI Tutor. The AI Tutor is launched from the main menu via the “AI Tutor” button.

Scene and Mode Management:
A scene/mode management layer is responsible for launching the appropriate Unity scene and configuring the active mode. When the student starts a module, the system loads the required 3D assets and prepares the UI for the selected mode (learning vs. review). This separation prevents quiz and AI features from coupling tightly with the main exploration flow.

Interaction and Selection Handling:
The interaction subsystem manages core VR interactions such as grabbing, rotating, releasing, and selecting anatomical structures. It also triggers visual feedback (e.g., highlight) when a structure is selected. The selection output is used by the UI to show labels and short info cards, and by content lookup logic to fetch the correct text entry.

Local Content Layer (Assets + JSON):
The application ships with a local content package that includes 3D model assets and JSON files. These JSON files contain structure labels/descriptions and the quiz question pool. The exported JSON is the runtime source of truth for learning text and quizzes, enabling offline use and keeping content changes independent from Unity scripts.

Content Loader:
A content loader reads the packaged JSON data and serves the appropriate label and short description for a selected structure. The loader uses stable identifiers/tags (e.g., module_tag for scope and structure_id for individual objects) to perform a simple lookup and return the correct UI text.

Quiz Manager:
The quiz manager loads questions from the local JSON pool, displays them in VR,

processes student answers, and shows immediate feedback and results. Quiz mode is intentionally independent from module exploration, allowing it to be launched directly from the main menu without requiring a module to be active.

AI Tutor Client:
The AI Tutor client is a dedicated UI screen that allows the student to ask questions via typing or push-to-talk (speech-to-text). It sends the question to the backend and displays a short text answer. Optionally, the student can play the answer aloud using text-to-speech. In the current scope, the AI Tutor is not structure-aware and does not automatically incorporate the currently selected structure as context.

Azure AI Backend (RAG Service):
The backend receives AI Tutor questions and generates grounded answers using a Retrieval-Augmented Generation (RAG) approach. It retrieves relevant content snippets from an indexed knowledge store and then produces a concise response. If the backend is unavailable, the AI Tutor feature is disabled while the core VR exploration and quiz features remain functional.

Retrieval Index / Knowledge Store:
The retrieval layer stores the learning content (e.g., curated notes or course material) in a form suitable for search. It provides the most relevant text chunks to the RAG pipeline so that generated answers remain aligned with the project's educational content.

2.3 Key Data Flows

This subsection summarizes the main runtime flows of VR Anatomy. Each flow is explained from the initial user action to the visible outcome inside the headset. For each flow, an activity diagram is provided to show the control steps and the decision points.

2.3.1 Module Launch Flow

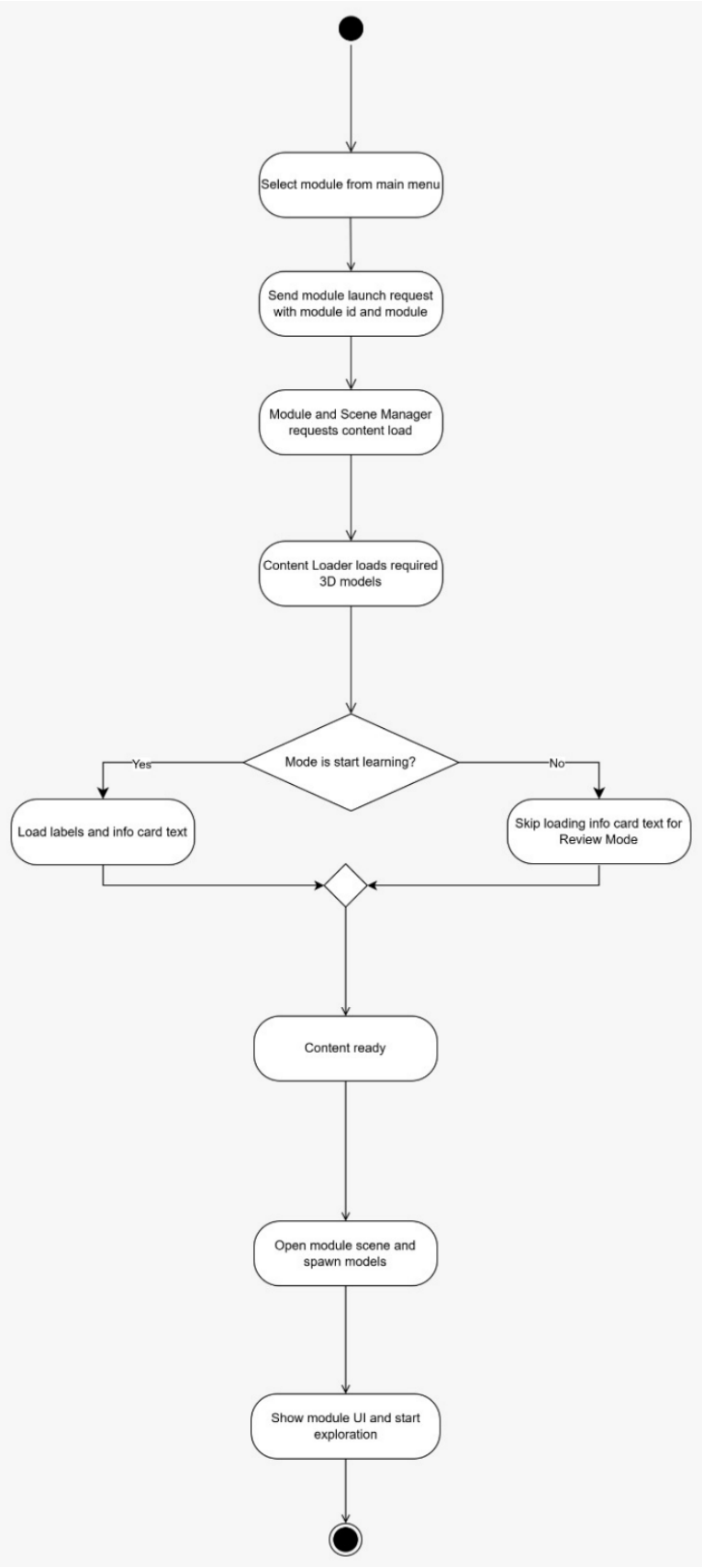

Figure 2. Activity diagram of the "Module Launch Flow".

2.3.2 Structure Selection and Info Card Flow

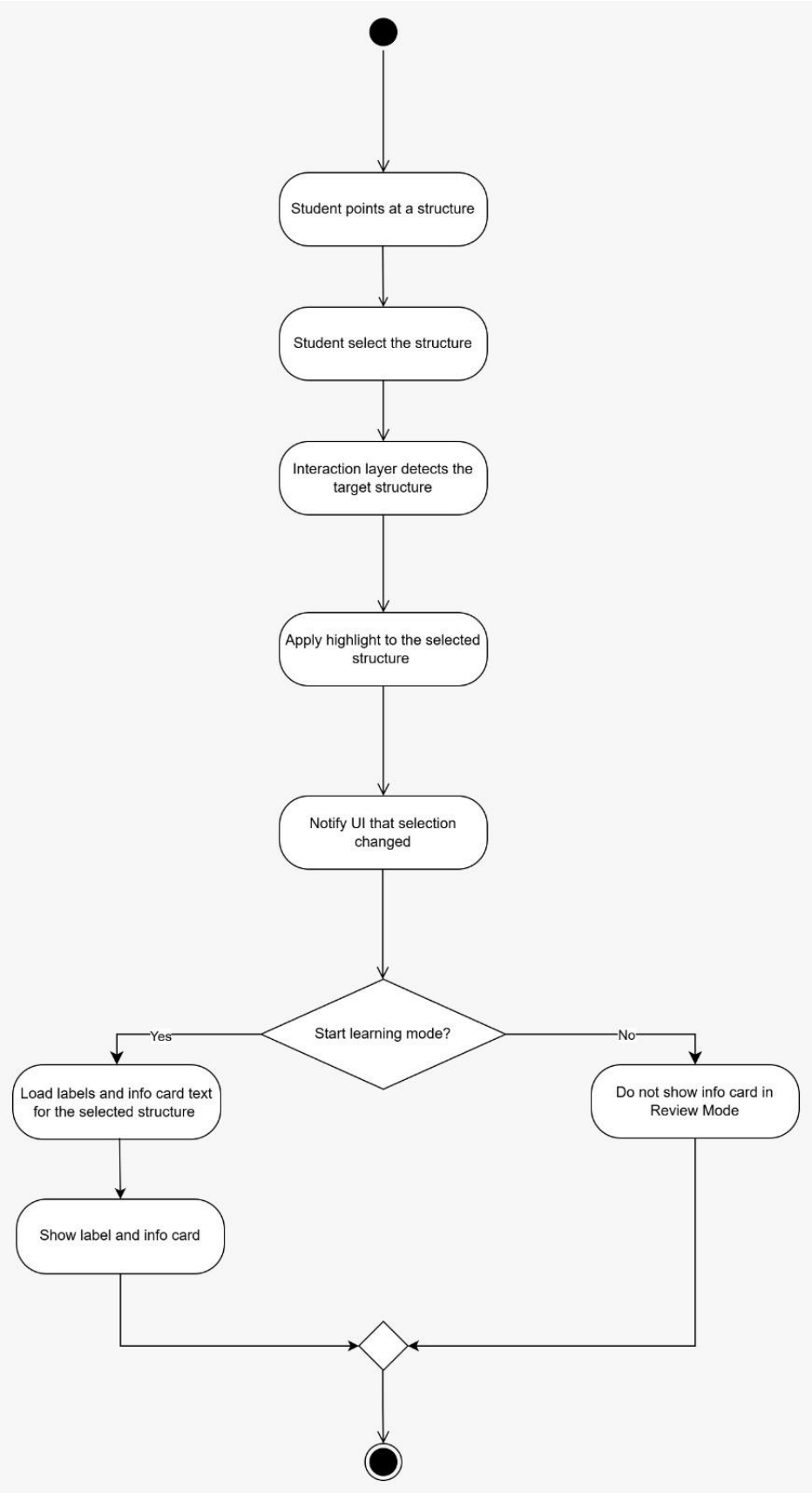

Figure 3. Activity diagram of the "Structure Selection and Info Card Flow".

2.3.3 Quiz Mode Flow

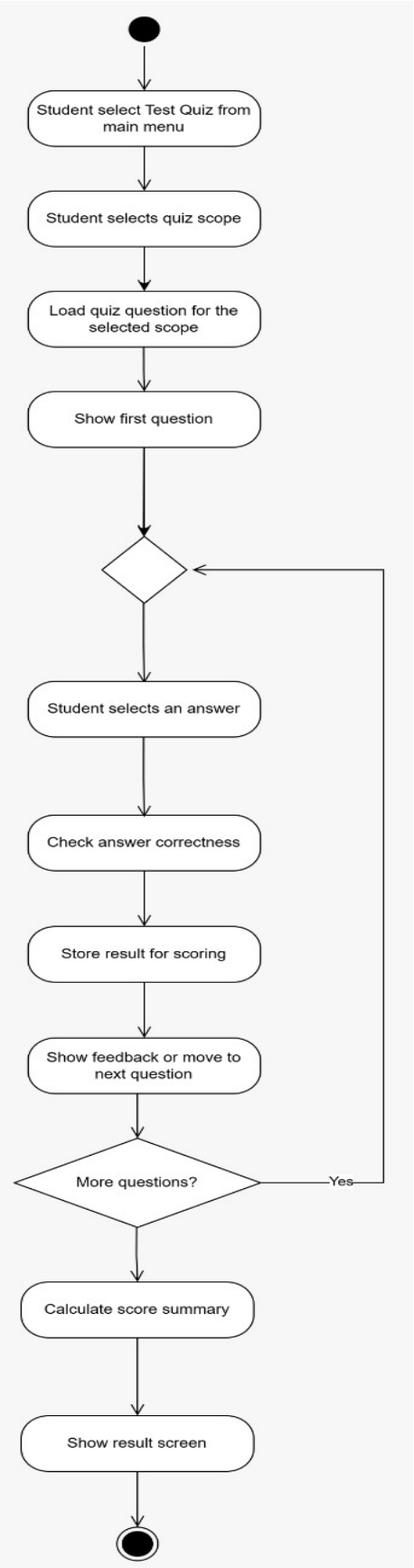

Figure 4. Activity diagram of the "Quiz Mode Flow".

2.3.4 Review Mode Flow

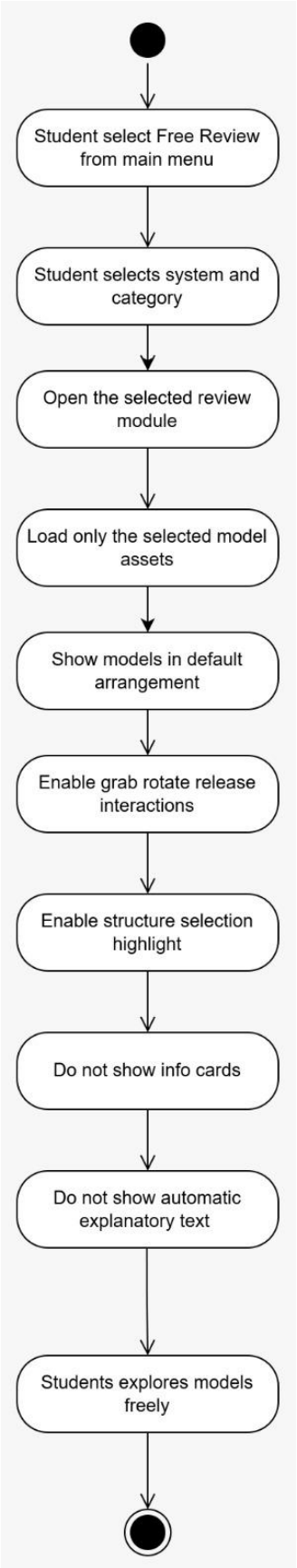

Figure 5. Activity diagram of the "Review Mode Flow".

2.3.5 AI Tutor Chat Flow

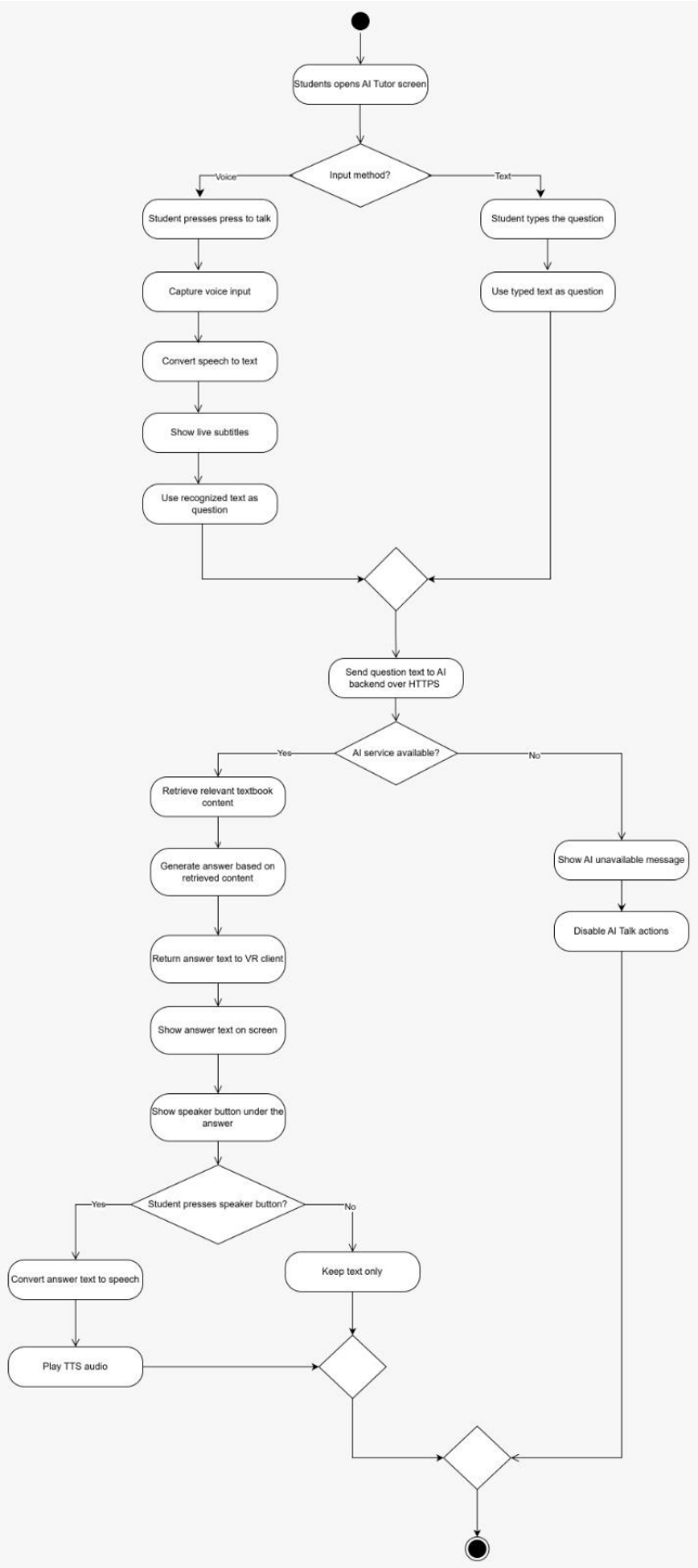

Figure 6. Activity diagram of the "AI Tutor Chat Flow".

3. Module Design in Unity

This section explains how VR Anatomy is organized inside Unity with the current main menu structure. It describes scene and mode organization, interaction handling, user interface design, content mapping, and fallback behaviors that keep the experience usable.

3.1 Scene and Module Structure

The Unity project is organized around a main menu entry scene and a small set of mode scenes or mode states. The main menu provides access to learning, free review, quiz, settings, AI tutor, and about screens.

A practical scene and mode structure is listed below:

- Main Menu scene as the entry point
- Learning mode scenes for module based exploration
- Free Review mode scenes that load selected model assets without info cards
- Quiz mode scene or UI flow that is independent from learning and review
- AI Tutor scene that opens as a full screen chat and voice interface
- Settings scene or panel that is reachable from the main menu
- About scene that shows project summary and credits

The main menu includes a hierarchical navigation structure for learning and free review. This navigation determines which content and model assets are loaded.

Start Learning supports guided exploration using hierarchical category selection. The student first selects an anatomy system and then selects a category within that system. In the current scope, both the Musculoskeletal System and the Circulatory System follow the same menu structure. The content coverage may be expanded in later iterations as new model assets and JSON entries are added.

Free Review uses the same category hierarchy as Start Learning. The difference is the mode behavior. Free Review loads only the selected model assets and does not show info cards or automatic explanatory text.

Test Quiz is accessed from the main menu and is independent from the category navigation used in learning and review. It provides three entry options for question pools. Musculoskeletal System, Circulatory System, and All Questions.

Settings, AI Tutor, and About are standalone screens that are reached directly from the main menu. They do not use the anatomy category hierarchy.

3.2 Interaction Layer

Key responsibilities are listed below:
• Detect structure selection using ray targeting or direct hand targeting
• Apply visual highlight to the selected structure
• Trigger label display and info card display in learning mode
• Support grab, rotate, and release interactions
• Provide a reset action on the controller that clears the current selection and restores the module to its initial state

The reset action supports classroom use. After the student moves or inspects parts, a single button press returns the model arrangement to the default state seen when the module first opened. This reduces confusion and helps the instructor quickly continue the lesson.

3.3 UI Layer

The UI layer supports navigation and learning feedback inside VR. It includes the main menu, mode entry screens, and in scene feedback such as labels and short info cards. The UI is designed to remain readable and to avoid covering the 3D model during exploration.

The UI is organized into four parts. These are:
• Main menu and global screens
• In module UI for learning and free review
• Quiz UI
• AI Tutor UI

3.3.1 Main menu and global screens

The main menu is the entry point of the application. It provides access to Start Learning, Free Review, Test Quiz, Settings, AI Tutor, and About. Start Learning and Free Review use hierarchical category selection to determine which module content and model assets are loaded.

Settings is a global screen intended for configuration such as sound related options. The exact set of settings may be expanded later. About is a global screen that presents project summary, team information, contributions and versions.

3.3.2 In module UI for learning and free review

In module UI supports exploration while the student interacts with 3D models. Learning mode shows labels and short info cards for the selected structure. Free Review follows the same exploration flow but does not show info cards or automatic explanatory text.

In module UI elements are listed below:
• A label that shows the selected structure name

• A short info card in learning mode that shows a brief description
• A minimal navigation element to return to the main menu
• A reset control that restores the module to its initial state

3.3.3 Quiz UI

Quiz UI is separate from modules and is reached from the main menu. It is designed for clear question presentation and quick answering.

Quiz UI elements are listed below:
• Question text area
• Answer options with selection feedback
• Result feedback such as correct or incorrect
• Controls for next question and exit to the main menu

3.3.4 AI Tutor UI

AI Tutor opens as a dedicated screen from the main menu. It supports voice-based interaction while showing text to keep answers clear in a classroom environment.

AI Tutor UI elements are listed below:
• Press to talk input control
• Live subtitles that show speech recognition output
• A control to play the answer using text to speech when available
• A clear unavailable state when the AI service cannot be reached

When the AI service is unavailable, the UI disables input and shows a short message. Core VR features remain usable.

3.4 Content Mapping

Content mapping connects three elements. The main menu selections, the 3D objects in Unity, and the local JSON content packaged with the application. The goal is to show the correct label and short description when a structure is selected in learning mode. The same mapping also supports quiz question selection by system and by all questions.

A minimal mapping approach uses stable identifiers.
• Each category that can be opened from Start Learning or Free Review has a module_tag
• Each selectable object inside a module has a structure_id
• JSON files store structure descriptions keyed by module_tag and structure_id
• Quiz question pools are stored keyed by system scope and may also support a combined pool for all questions

• Unity objects store identifiers through a small component so runtime lookup is simple

At runtime, the mapping follows a clear sequence.
• The student selects a category from the main menu
• The VR client loads the corresponding module content and model assets
• The content loader reads the JSON entries for that module_tag
• When the student selects a structure, the VR client uses structure_id to request label and description data
• The UI layer shows the label and short info card in learning mode

Free Review uses the same module and structure identifiers but does not display info cards. The mapping still supports selection highlighting and label display.

Quiz mode uses a different mapping entry point. The student chooses a question pool option in the quiz screen. The VR client then loads the related pool from JSON. The All Questions option merges or selects a combined pool defined in the content package.

Module and structure identifiers are assigned and maintained during content preparation. The same identifiers are stored in JSON and in Unity objects, which keeps content mapping consistent for the current project scope.

3.5 Error Handling and Fallback Behaviors

The Unity client should remain usable when some content is missing or when external services are unavailable. Fallback behaviors are designed to keep the student in a valid learning state and to avoid blocking core exploration.

Error handling follows two principles:
• Fail safely by disabling only the affected feature
• Communicate clearly using short, non-technical messages in VR

Typical cases and expected behaviors are listed below:
• If a structure has no JSON description, the client still allows selection and highlighting, and shows only the label
• If a module content package is incomplete, the client blocks entry to that category and returns to the main menu with a short message
• If a quiz pool is missing for a selected option, the client disables that option or shows a message and returns to the quiz selection screen
• If the AI service is unreachable, the client shows an unavailable state and disables AI input, while learning, free review, and quiz remain usable
• If the AI response is slow, the client shows a loading state and applies a timeout, then shows a short message
• If speech recognition or text to speech is unavailable, the client keeps the text based interaction when possible and shows that voice features are unavailable

During development, errors should be logged consistently so the team can reproduce issues. During classroom use, messages should remain short and focused on the next action the student can take.

4. API Design

This section describes the minimal API between the Unity VR client and the AI backend on Azure. The API is used only for the AI Tutor feature. Core VR features such as module exploration, selection highlighting, info cards, and quiz use local content and do not require the API.

The API is designed to be simple, stable, and easy to test. It uses HTTP over TLS and JSON messages. The client sends a question with module context, and the backend returns a grounded answer based on project provided learning content.

4.1 API Overview

The AI Tutor feature uses one endpoint. This endpoint accepts a question and returns a grounded answer. The backend uses a retrieval step over project provided learning content and then produces a response based on the retrieved text. This is the RAG approach in this project. It helps the system answer using the provided material instead of guessing from general knowledge. The AI Tutor is not structure aware. The request does not include selected structure fields.

4.2 AI Tutor Endpoints

The AI Tutor feature uses a minimal REST endpoint.

The Unity client will call this endpoint after integration is completed. The Azure based RAG system is already operational through a web interface using an anatomy textbook as the primary content source. The purpose of the endpoint is to accept a student question and return an answer grounded in the project learning content.

The AI Tutor is not structure aware. The request does not include selected structure information.

4.2.1 Request Schema

The request body is JSON. The schema is described as a planned contract for the Unity client.

The only strictly required input for answering is the student question. Additional fields may be added to support integration, debugging, and content scoping.

Recommended minimal field is listed below:
• question as a string and required to generate an answer

Recommended fields that may be used during Unity integration are listed below:
• session_id as a string to trace requests and support debugging
• module_tag as a string to limit retrieval to a selected scope if the content is later split by systems or categories

Design notes are listed below:
• If one textbook is used as a single content source, module_tag may be omitted
• If multiple content scopes are introduced later, module_tag helps keep answers relevant
• session_id is not required for answering but helps identify which Unity session produced a request

4.2.2 Response Schema

The response body is JSON. The schema is described as a planned contract for the Unity client because Unity integration is a later step.

The main goal of the response is to return a short answer that can be displayed comfortably in VR. The response may also include supporting references when the backend can provide them.

Recommended minimal fields are listed below:
• answer as a string
• session_id as a string

Optional candidate fields are listed below:
• citations as a list of objects with short supporting snippets
• meta as a short object for non-personal diagnostics such as latency

Design notes are listed below:
• citations are useful to increase trust because they show a brief supporting excerpt from the learning content
• if citations are not available in the current backend setup, the response may contain only answer
• session_id helps the client match the response to the request during integration testing

4.2.3 Error Responses and Timeouts

The AI Tutor endpoint must fail safely because the core VR learning experience should remain usable even when the online service is unavailable. Error handling is designed to be predictable for the client and understandable for the user.

General principles:

- The VR client treats the AI Tutor as an optional feature.
- When an AI request fails, the client keeps the current module state unchanged.
- The client shows a short message that explains that the AI Tutor is unavailable and invites the user to try again.
- Errors are logged for debugging, but logs must avoid personal data.

Planned error response shape:

- error_code as a short stable identifier
- message as a user safe explanation
- session_id when provided in the request
- retry_after_seconds when a retry delay is recommended

Expected error categories:

- Invalid request: Returned when required fields are missing or values are not valid.
- Authentication or authorization failure: Returned if the endpoint later requires keys or tokens.
- Rate limiting: Returned when the service is overloaded or a client exceeds allowed request volume.
- Upstream AI failure: Returned when the hosted AI pipeline cannot produce an answer due to temporary issues.
- Internal server error: Returned for unexpected failures

Timeout policy: The client should use time limits to avoid blocking the VR experience.

- Connection timeout is short to detect offline states quickly.
- Total request timeout is moderate to allow retrieval and generation, but still protects interactivity.
- If a timeout happens, the client may allow a manual retry rather than automatic repeated retries.

Client behavior on failure:

- Disable or gray out AI Tutor entry points when the service is unreachable.
- Keep main menu, module browsing, selection, labels, and quiz features available.
- Avoid repeated popups. Prefer a single banner style message and an explicit retry action.

4.3 Security and Privacy Considerations

VR Anatomy is a single user VR application with an optional online AI Tutor. Security and privacy work focuses on reducing risk while keeping the system simple for an academic project.

Data minimization:

- The AI request should include only what is needed to answer the question.

- The system should not send selected structure identifiers because the AI Tutor is not structure aware.
- The system should avoid sending user identifiers. A short session_id may be used only for matching requests and responses.

Personal data handling:

- The VR application does not need personal profiles or accounts for the current scope.
- Voice input may contain sensitive information. The UI should warn users not to share personal data.
- If speech features are enabled in AI Tutor, audio should be treated as transient input and not stored by the VR client.

Transport and endpoint protection:

- If an API key or token is required in the future, it should not be hard coded in the Unity client.

Logging and diagnostics:

- Logs should help debugging without exposing user content.
- By default, logs should not store full questions or audio transcripts.
- If limited logging is needed during testing, it should be time bounded and access controlled.

Abuse and reliability:

- Basic rate limiting may be applied on the server side to protect availability.
- The client should implement safe retries with backoff and stop after a small number of attempts.
- When AI is unavailable, the client should disable AI Tutor while keeping core learning and quiz features available.

Content safety boundaries:

- The AI Tutor is intended for anatomy learning questions that relate to the provided textbook.
- The UI should set expectations that answers may be incomplete or wrong.
- The system should encourage students to verify critical information with course material or an instructor.

5 Deployment and Runtime Environment

This section describes how VR Anatomy runs during a typical session and how the optional AI Tutor fits into the runtime. The goal is a stable classroom style setup with a single headset and a single student at a time.

5.1 Runtime Setup and Execution Context

VR Anatomy runs as a Unity based VR application on a single headset. One student uses the headset at a time. The instructor is a secondary user who guides verbally and does not interact with the application directly.

Runtime characteristics:

- The core learning experience runs locally on the headset.
- Learning modules load 3D models from Unity assets.
- Labels and short info cards use local text content.
- Review Mode uses the same model assets but does not show info cards or automatic explanatory text.
- Quiz Mode is started from the main menu and runs independently from learning modules.

The AI Tutor is an optional feature accessed through the AI Tutor screen. When available, it connects to an Azure hosted backend over the network. If the AI service cannot be reached, AI Tutor is disabled while the rest of the VR application remains usable.

5.2 Offline and Online Operation

VR Anatomy is designed so that learning modules work without online services. Online access only improves the experience through AI Tutor.

Offline operation:

- Start Learning modules are available with 3D exploration and basic interactions.
- Selecting a structure still highlights it.
- Labels and short info cards still appear for selected structures.
- Review Mode works and keeps its rule set with no info cards or automatic text.
- Quiz Mode works as long as quiz data is available locally for the build used in testing.

Online operation:

- All offline features remain available.
- AI Tutor becomes available when the AI backend can be reached.
- If the AI service fails during a session, AI Tutor is disabled and the rest of the application continues.

5.3 Dependencies and Configuration

The project uses a small set of dependencies to keep the build stable and easier to test.

VR client dependencies:

- Unity project with VR interaction support

- XR plugins and device runtime required by the selected headset
- 3D model assets included as Unity assets
- UI assets for menus, labels, info cards, and quiz screens

AI Tutor dependencies:

- An Azure hosted backend that exposes the AI answer endpoint
- A retrieval setup that uses the indexed textbook content
- A language model service used by the backend

Configuration approach:

- VR client settings are stored in Unity project configuration and prefabs.
- Settings menu includes sound related options that are planned.
- AI endpoint base URL should be configurable and not hard coded for long term use.
- Secrets such as keys should not be stored in the VR client. They should be handled on the server side.
- Quiz and structure descriptions are planned to be exported to JSON later. This packaging is not finalized yet.

6 Design Decisions

This section explains the main design choices behind VR Anatomy. Decisions were made to support a stable VR learning experience, limit development risk, and keep the system testable within a graduation project scope.

The project prioritizes:

- Reliable core learning features that work without online services
- Simple interactions that students can learn quickly
- A modular structure that supports iterative content growth
- Clear separation between VR learning and optional AI support

6.1 Why Module Based Scenes

The application uses module based scenes to keep content isolated and easier to manage.

Key reasons:

- Each module can be loaded and tested independently.
- Large 3D assets stay inside the relevant module and do not affect other modules.
- Scene separation supports parallel work. One team can work on UI while another team improves a specific module.
- Memory usage is easier to control because only the active module needs to be loaded.

- The reset button behavior is easier to implement. Reset returns the module to the initial state by clearing selection and restoring default arrangement.

6.2 Why JSON for Content

The project stores learning text and quiz content in JSON files to separate content from Unity code. These JSON files are bundled with the VR application and serve as the runtime source of truth for labels, short descriptions, and the quiz pool.

Expected benefits:
• Content can be updated without changing Unity scripts.
• Structure descriptions and quiz questions follow a consistent format.
• Core learning and quiz features can work offline using the packaged JSON content.

Status and limits:
• During development, quiz questions may be authored in an Excel file for convenience, but the VR client uses exported JSON at runtime.
• JSON files are shipped with the app as part of the local content package.
• 3D models remain Unity assets and are not part of the JSON content.

6.3 Why Review Mode Without Info Cards

Review Mode removes info cards and automatic explanatory text to support focused visual learning.

Rationale:

- It encourages students to rely on recognition and exploration rather than reading.
- It supports instructor guided sessions where explanations are given verbally.
- It reduces visual clutter and keeps the scene clean for examination.
- It separates learning from assessment. The same model can be reviewed without hints.

Resulting behavior:

- Only the selected model assets are shown.
- Selection highlighting and basic grab and rotate interactions remain available.
- No info cards appear after selection.

6.4 Why AI Tutor Is Optional and Not Structure Aware

The AI Tutor is optional because the core VR experience must remain usable offline and without external dependencies. The AI pipeline currently works through a web interface, while Unity integration is planned later.

Reasons for optional integration:

- Classroom use must not fail when the network is unstable.

- Testing and demos remain possible without relying on online services.
- The system scope stays realistic for a graduation project.

Reasons for not being structure aware:

- The current AI setup is based on the textbook ingestion and retrieval flow.
- Passing selected structure identifiers would suggest precision that the system does not reliably support.
- Keeping requests minimal reduces coupling between VR interaction state and the AI backend.

User experience rule:

- If the AI service is not available, AI Tutor is disabled while modules and quiz remain available.

6.5 Performance and VR Comfort Decisions

VR comfort and performance are treated as core quality goals because the application runs on a headset and uses real time interaction.

Performance decisions:

- Keep scenes simple and avoid loading unnecessary assets outside the active module.
- Prefer optimized 3D models suitable for real time VR.
- Use stable lighting choices where possible and avoid expensive effects during interaction.
- Limit the number of on screen UI elements during exploration.

Comfort decisions:

- Avoid forced camera movement and sudden viewpoint changes.
- Prefer direct object interaction such as grab, rotate, release.
- Provide the reset button to quickly return to a known safe state.
- Keep UI readable and consistent in size and placement.

7 Limitations and Risks

This section lists the main limitations and risks of VR Anatomy within the current project scope. The goal is to make constraints explicit and define what may affect stability, learning and quality.

7.1 Technical Limitations

The project targets a single headset and a single user at a time. This keeps the system simple but limits usage scenarios.

Main limitations:

- No multiuser support and no shared session features.
- No instructor control panel. The instructor guides verbally outside the system.
- Limited hardware coverage. The project is tested on the chosen headset and configuration.
- Module content depends on Unity assets. Large assets may increase load times and memory use.

Practical impact:

- The application is suitable for guided demos and small classroom sessions.
- Porting to other devices may require extra XR configuration and performance tuning.

7.2 AI Related Risks

The AI Tutor improves access to explanations, but it introduces risks because it depends on online services and generated text.

Main risks:

- Availability risk. Network or backend issues can disable AI Tutor.
- Answer quality risk. Generated answers may be incorrect, incomplete, or misleading.
- Context limitation. The AI Tutor is not structure aware, so it may not understand what the student is looking at in VR.
- Safety and privacy risk. Voice input may include personal data if users speak freely.

Mitigation approach:

- Keep AI optional and ensure VR modules and quiz work without it.
- Keep requests minimal and avoid sending structure identifiers.
- Avoid storing voice or question content in logs by default.

8 Conclusion

VR Anatomy is a Unity based VR learning application that supports interactive anatomy exploration for a single student using a single headset. The system provides module selection from a main menu, 3D exploration, structure highlighting, basic grab and rotate interactions, labels, and short info cards. Quiz Mode is available independently from modules, while Review Mode offers a clean study view without info cards or automatic explanatory text. A reset action restores each module to its initial state to support repeatable practice.

The architecture cleanly separates core VR learning features from online AI support. The AI Tutor provides textbook grounded help for student questions using retrieval and answer generation. The AI component is optional. This ensures that modules, labels, info cards,

Review Mode, and Quiz Mode continue to work even when network access or the AI service is unavailable.

This design keeps the project realistic for a graduation scope. It supports iterative content growth, stable classroom demos, and clear boundaries between interactive learning, review, assessment, and optional AI assistance.