VIRA | Virtual Insight into Real Athletics

Design Document

Group 33

Advisor

Dr. Diane Rover

Team Members

Katie Perkins	*Team Lead*
Nate Irmiter	*Stakeholder Communicator*
Bailey Righi	*Product Manager*
Willem Paul	*Standardization Manager*
Caroline Rankin	*Meeting Facilitator*

Email

sdmay20-33@iastate.edu

Website

sdmay20-33.sd.ece.iastate.edu

Revised

Sunday, December 8, 2019
Version 3

Executive Summary

Development Standards and Practices Used

Software Development Practices

- Agile development process
- High cohesion, low coupling
- Modular code for maintainability and extensibility
- Good code formatting and documentation

Engineering Standards

- Unit testing
- Integration testing
- Acceptance testing
- Hardware I/O testing
- Documentation of design decisions, including UI and architecture
- Documentation of technologies and rationale behind them
- IEEE Standards
- Oculus application development standards

Summary of Requirements

- The user can log in and navigate to a guided special test module or injury evaluation quiz module
- The user can view the virtual limb in skin, muscle, or tendon modes to aid them in identifying what muscles and tendons lie underneath the skin
- The user can see their performance data from the modules they have worked on in the form of a performance dashboard

Applicable Courses from Iowa State University Curriculum

We have all taken different courses, so every group members' experience is different. In general, however, the list of applicable courses includes (but is not limited to):

S E/CPR E 185	*Problem Solving in Software Engineering*
COM S 227	*Object-Oriented Programming*
COM S 228	*Introduction to Data Structures*
S E/COM S 309	*Software Development Practices*
S E 319	*Construction of User Interfaces*
S E 329	*Software Project Management*
S E 339	*Software Architecture and Design*
S E/COM S 362	*Object-Oriented Analysis and Design*

New Skills/Knowledge Acquired Not Taught in Courses

- Unity development environment
- Virtual reality application design
- Computer graphics concepts and terminology
- Knowledge of human anatomy and medical diagnosis procedures
- Better understanding of software development processes
- Creation of human models with MakeHuman
- Model development and animation with Blender
- Knowledge of software engineering and physical therapy standards

Table of Contents

1 | Introduction

2 | Specifications and Analysis

3 | Statement of Work

4 | Project Timeline, Estimated Resources, and Challenges

5 | Testing and Implementation

List of Figures

List of Tables

1 | Introduction

1.1 | Acknowledgement

We would like to acknowledge the significant help of Joe Hubbard, an athletic trainer at Iowa State who is our primary source of information related to athletic injury examinations. We would also like to thank Dr. Diane Rover for her guidance and expertise throughout the duration of our Senior Design project.

1.2 | Problem and Project Statement

There is nothing as good as hands-on, in-person, real-time experience in performing the tasks of an athletic trainer. When students in Iowa State's athletic training program have made it to the point at which they are able to work in the training room with student athletes, their time there becomes very valuable. However, while they are learning, a full-time athletic trainer must watch them and guide them through what they should do. This makes it very difficult for athletes to get the attention they need when the training room gets busy. On top of that, there are many traumatic injuries that a student will never experience firsthand until they are working full-time in the field.

A virtual reality (VR) training program would allow athletic training students to get the extra experience and feedback they need without having to interrupt or wait for a full-time athletic trainer to finish treating another athlete. We propose a virtual reality program that consists of a set of training modules that walk an athletic training student through injury evaluation and diagnosis scenarios.

Our goals are as follows:

1. Create a virtual reality training program that provides student athletic trainers with more experience outside of class and work.
2. Give student athletic trainers realistic simulations of diagnosing injuries without performing real (and critical) examinations.
3. Allow for easy future extension of VIRA by using a modular design.

1.3 | Operational Environment

VIRA is primarily designed for stationary, indoor use. The VR headset we have chosen to use—an Oculus Quest—should therefore not be subjected to extreme conditions. As with any VR application, users should be aware of their surroundings at all times. The Oculus Quest has a built-in feature called the "Guardian", a virtual box around the user that alerts them when they step or reach outside of its bounds. This further protects the user and the headset from injury or damage.

1.4 | Requirements

Functional Requirements

Before the Module

- The user can select a user profile when starting the application
- The user can select a mode (guided or quiz)
- The user can select a specific module
  - Guided Mode modules will focus on special tests
  - Quiz Mode modules will focus on injury evaluation scenarios
  - Modules will cover traumatic injuries not often seen in training

During the Module

- The user can toggle view modes to see the skin, muscles, or tendons of the virtual limb
- The user can select a muscle or tendon and see its name
- The user can manipulate the limb in a realistic manner
- The application can guide the user through a special test
- The application can assess the user on an injury evaluation quiz
  - The user loses points for the following:
    - assuming the injury before evaluating,
    - palpating the wrong spot(s) or not palpating a certain spot,
    - doing incorrect or unnecessary tasks,
    - giving an incorrect diagnosis,
    - completing poor differential diagnoses, or
    - skipping steps of an evaluation
- The user can view the angle at which they are bending a joint when evaluating range of motion
- The user can exit the module and return to it at any time

After the Module

- The user can see their progress on a particular module over time
- The user can restart the module or return to the menu

Non-Functional Requirements

- The application must respond to user input in real time
- The user must feel comfortable while using the application
  - The application must not induce motion sickness or nausea in most users
- The device must have a battery large enough to operate during a half-hour session

Economic Requirements

- We shall not exceed the funding provided to us by the Department of Electrical and Computer Engineering (hereinafter “the Department” or “ECpE”)

Environmental Requirements

- The application must work in real time and interact directly with the user
- The application must accurately interpret the motion of the user’s hands and realistically translate that motion to the virtual display
- The user must remain within the Guardian that is free of obstacles and other people while VIRA is in use

1.5 | Intended Users and Uses

VIRA is intended for use by student athletic trainers to practice injury evaluation and diagnosis. These are students who wish to further develop their clinical skills and gain experience working with traumatic injuries they would not typically see as a student worker. Our intended customer is the Iowa State Department of Sports Medicine, as they will be assisting us with module testing and will evaluate our accuracy. Our goal is to create an educational program consisting of modules that will augment students’ studies and on-the-job experience.

1.6 | Assumptions and Limitations

Assumptions

For this project, we make the following assumptions:

- The Oculus Quest will allow a great enough range of motion to simulate interaction with an athlete.
- The Oculus Quest has a sufficient level of hand-tracking capabilities for VIRA.
- The maximum number of users will be at one at any given time.
- Multiple users will share the same device.
- The end product will not be used by anyone but ISU athletic training students or their mentors.
- The users have little experience in the specific diagnosis module(s) we create.
- The users have a sufficient understanding of English.
- The users are physically capable of operating an Oculus Quest.

Limitations

As VIRA is a student-proposed project and due to complications acquiring hardware, our limitations are as follows:

- We are using an Oculus Quest, restricting us to VR technology.

- None of us have prior experience in VR development.
- We do not have corporate funding and are therefore operating on a small budget from the Department.
- The number of VIRA modules created will be limited by development time and any schedule setbacks.
- The quality of the graphics will be limited by the version of Unity we are able to use and the functionalities available to us in Unity as well as the quality of any other graphics programs we choose to utilize.
- VIRA requires an internet connection and can therefore only be used in a location with a network signal.

1.7 | Expected End Product Deliverables

VR Training Module

The final product will consist of at least one VR training module by the end of the school year. This will include both the standard training mode, in which the user is guided through a specific injury evaluation and provided with feedback (see below), as well as a quiz mode, in which the user will be assessed on their performance. The specific skills targeted by these modules are special tests and injury evaluation or diagnosis.

Feedback Dashboard

The module(s) will implement a feedback component. This system will monitor the student athletic trainer's progress as they work through a module and will assess their technique, accuracy, etc. and provide them feedback after they complete the module. This information will be connected to a specific user's account so that user will be able to see what they did well and what they still need to improve upon.

2 | Specifications and Analysis

2.1 | Proposed Design

The program will be written for an Oculus Quest headset and controllers using Unity and will consist of modules that address scenarios that an athletic training student does not typically see while working in a clinic. The user will work through their chosen module alone, either standing still or sitting down. The module will consist of either a guided special test or a quiz evaluation of a virtual limb that is associated with the chosen scenario. The user will be able to manipulate the limb and gather more information by seeing different views of it, such as the muscle layer or tendon layer. The latter two enable further practice in palpation, which is key for injury evaluation.

For a guided special test, the program will prompt the user with visual cues to complete the evaluation process. For a quiz evaluation, there will be no guides for the reader, and they must manipulate the limb and make decisions about the options presented to them on their own. At the end of the training module, a virtual dashboard will appear to give the trainer feedback on how they performed during the training. It will also keep a history of how they have done on that particular module in the past so they can see the progress they are making towards doing it correctly. This project is intended to be scalable due to how unfamiliar the team is with virtual reality. We will begin with a single training module, then create others if time allows.

To test our modules, we plan on following IEEE 29119 standards for testing processes and techniques. We plan to utilize GitLab's continuous integration capabilities and in doing so, we will be organizing our test plan development, documenting, and discussing our plan. Additionally, many aspects of identifying, estimating, and handling risks have been a part of our design process, so we have been and will continue to take these steps into consideration. As for testing techniques, we plan to design our modules in a manner that heavily controls the program flow since VIRA is based on strictly defined injury tests. As such, there are some IEEE standards we plan to follow [1]. Most of the structure-based techniques will have a direct application to our modules and others will be very useful, such as boundary value analysis—we want to ensure that our models move correctly and have bounds on their movement—and state transition testing—we want to ensure that specific program or modules states flow to their correct successors.

Our project must also be compliant with the Oculus Quest Virtual Reality Check (VRC) Guidelines [2]. These detail the packaging, audio, performance, functional, and security requirements, among others. All applications submitted to the Oculus Store must comply with these requirements and we intend to develop VIRA as if we were submitting it to the Store.

Additionally, we plan on developing our project according to a few different IEEE virtual reality standards. The first of these is "P2048.5 – Standard for Virtual Reality and Augmented Reality: Environmental Safety" [3]. Since our project has the goal of helping with injury, the last thing we want is for our users to injure themselves with using our product. To mitigate this, we plan on keeping safety in mind while designing VIRA's controls so that the users are not required to make any large, sweeping movements. This will help the Oculus Guardian fulfill its task of warning the users before they accidentally hit something or someone nearby. We also have the option to include a warning in VIRA as an extra precaution.

The second standard we will consider is "P2048.6 – Standard for Virtual Reality and Augmented Reality: Immersive User Interface" [4]. The whole point of using VR for our project is to be able to complete the tutorials and act in a physical manner that is similar to performing the assessments on a real patient. There is a large difference between keyboard and mouse controls and VR controllers. Because of this, as we design VIRA's controls, we need to ensure that the users are performing actions in the most intuitive manner possible so that they feel truly immersed in the experience.

2.2 | Design Analysis

We collectively decided that our process would follow the design thinking process presented in class. *Figure 1* (right) shows a diagram of our process, which we will repeat in iterations. In some ways, it has worked well so far, in others, we have found some flaws.

One issue we have discovered is that a complication in one area of the project has the potential to complicate another area. For instance, we met with an athletic trainer with the idea of creating an AR application, but due to difficulties acquiring a Microsoft HoloLens, we will instead be creating a VR application. This will have an impact on the potential functionality of the modules and as a result, on the information we need to gather from the trainer. It will likely manifest once we start development as we reconsider some of our planned functionalities. Currently, we do not plan to change our design process.

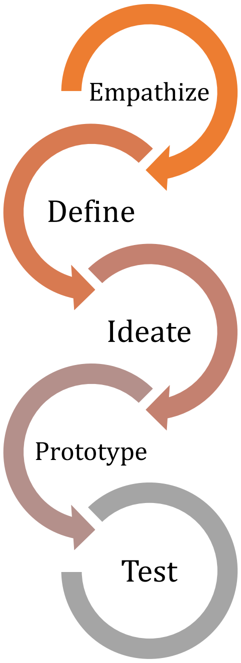


Figure 1 – Design Thinking Process

As we began prototyping VIRA, we began to see how our design choices have affected our workflow. We have decided that the best method for creating our humanoid models is to use the MakeHuman and Blender programs, then import them into Unity so that the models will look realistic and are easy to manipulate. We also attempted to find some models online that might be more anatomically realistic, but those models were either too complex to animate or were very expensive (on the order of hundreds of dollars). Using the MakeHuman models gives us the capability to use humanlike “athletes” without paying anything. All of these programs are designed to function with one another, which makes creating models and transferring them relatively easy. However, figuring out the process of creating the models, rigging and animating them, and transferring them between programs has taken a lot of time and there are still some aspects we find difficult. Another downside is that we will have to do a lot of research into animating the models the way we want. There will be difficulties in this area because animation in VR typically involves handling entire objects instead of individual parts, so creating natural-looking, realistic animations for our humanoid models will take a lot of trial and error.

2.3 | Development Process

We will use an Agile process to develop VIRA. It is crucial that we receive lots of feedback from the end-users to ensure that we create exactly what they need. Working iteratively will allow us to check in with them regularly and will give us a good structure for our own time management. An Agile process will also ensure that we work collaboratively to have a shippable product at the end of each sprint, rather than working individually and hoping that one member’s code will integrate with another’s.

2.4 | Design Plan

Figure 2 (right) shows our system component diagram, which we will use to guide our design. This diagram shows the different components of the system and how they communicate with one another.

The program flow begins with the GUI component. As the user interacts with the application, their input triggers events that are sent to the module manager component. This is our logic layer that runs in the background. It will handle all of the scene changes and program decisions, such as allowing the user to log in, select a module, work through a module, and view their performance. Both the GUI and the module manager components rely on the Unity Game Engine to function. The module manager also interacts with the

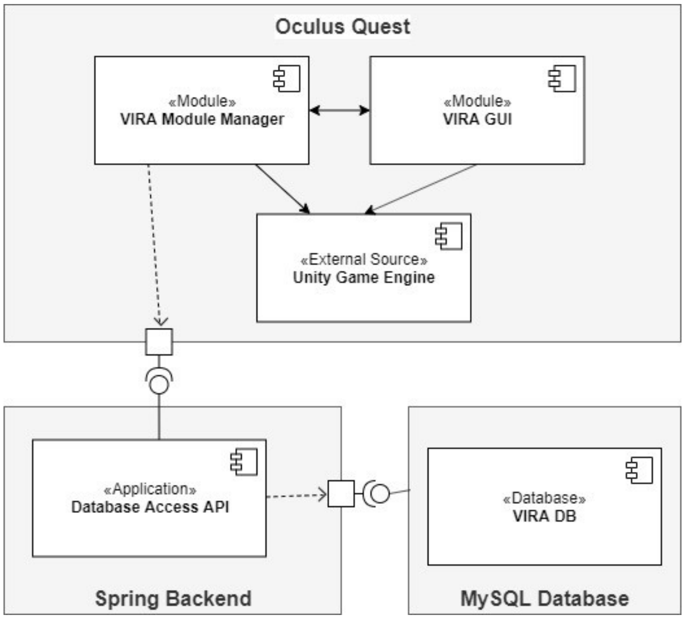


Figure 2 – VIRA Component Diagram

remote Spring server, which acts as an interface between the application and the database. In order to select a user profile, see user performance, save user performance, or see all modules, interaction with the database is required. Once the data is accessed, it will be cached on the Quest for the duration of the session for efficiency. We chose this method of accessing data because it is much more secure that accessing the database directly from the application and it allows us to complete the necessary calculations for the performance dashboard server-side, decreasing the amount of processing done on the Quest itself. As these user actions are the only ones that require persistent data, communication on the network is minimized.

3 | Statement of Work

3.1 | Previous Work and Literature

There are several existing simulations for medical procedures such as Touch Surgery, the Oxford Medical Simulation, Osso VR, and the Oculus-Children's Hospital Los Angeles collaboration. These products focus on creating a realistic simulation in which the user interacts with a virtual patient in a certain medical scenario. However, these scenarios are all procedures that require tools, as opposed to evaluations focused on the musculoskeletal system. These VR applications are organized into modules, which we would also like to do. Breaking such applications into modules helps users optimize their time and helps developers streamline development. The current products on the market today are able to map the human body and calculate its movements, which would also like to apply to VIRA. The biggest difference between these products and VIRA, aside from the targeted medical field, is that VIRA will implement manipulation of the human models. Our users will not only be able to interact with the virtual patient, but will be able to hold and move the patient's limbs. This is one of the greatest challenges we will face during development.

3.2 | Technology Considerations

During the proposal phase, we were considering two different routes for the projects: augmented reality (AR) or virtual reality (VR). We initially chose AR as the preferred technology as it would allow the user to practice injury evaluation with a real partner. They would be able to see another person and manipulate their joints. Unfortunately, after following several possible leads, we were unable to acquire a Microsoft HoloLens. As a result, we are instead using an Oculus Quest, a VR headset. VR technology, while still useful, may not provide the best experience for the user, but after some testing, we have determined that it should still provide us with sufficient capabilities to successfully implement VIRA. Despite this, there are tasks that an athletic trainer would perform while testing and diagnosing a patient that are difficult or impossible to simulate in a VR environment. For instance, testing a joint's range of motion would require the trainer to provide resistance to the limb. While there is no way to recreate this force in VR, we can simulate it using the Quest controllers' haptic feedback. This will still allow the user to learn the steps required to perform a certain test.

As for software, we plan on using MakeHuman and Blender to create our models. Using these in conjunction allows us to rig up humanoid models an import then into the Unity IDE, with the hope that the users will be able to manipulate the models. So far, we have figured out how to move the models around, but we still need to be able to move individual body parts in a realistic manner. It is unclear if the models were intended to be used in this way—for the most part, they are used as characters in non-interactive animations—so it will be a challenge to utilize these technologies for our purposes. Another potential downside is that we would like to reskin the models or use models with anatomically correct exteriors. Since we are working on a medical application, this would be a useful feature, but we will not focus on this as a priority right now since the medically accurate models we have found are not well-suited for animation.

3.3 | Task Decomposition

1. Create user profile selection
    a. Create screen sketches
    b. Develop basic graphics from screen sketches in Unity
    c. Code population of user profiles
    d. Code transition from user selection screen to user home screen
        i. Includes interaction with database
2. Create user home screen
    a. Create screen sketches
    b. Develop graphics from screen sketches in Unity
    c. Code transition from selecting an option to its respective scene
3. Create module selection
    a. Create screen sketches
    b. Develop basic graphics from screen sketches in Unity
    c. Code population of module options
    d. Code transition from module selection screen to module
4. Develop models for the chosen limb(s)
    a. Create different view modes
        i. Create skin view
        ii. Create muscle view
        iii. Create tendon/ligament view
5. Create guided special test module
    a. Establish narration/guided content with help from athletic trainer
    b. Establish statistics to collect for performance dashboard
    c. Create screen sketches
    d. Develop basic graphics from screen sketches in Unity
    e. Code ability for user to interact with and move a limb
    f. Code text and visuals for user guidance
        i. Limb movement indicators to guide user to move the limb in a certain way
        ii. Button selection for next step
    g. Code detection that the user has followed guidance correctly
        i. Detect if the user has moved the limb to the correct position as indicated (if applicable)

        ii. Code redirect in case the user has done something wrong (may include more specific guidance or more explanations)
        iii. Code statistics collection and store information in the database
    h. Code muscle selection and corresponding identification
    i. Code tendon/ligament selection and corresponding identification
    j. Code transition from module completion to performance dashboard
    k. Choose which sound clips to use and where to use them

6. Create quiz module
    a. Establish narration with help from athletic trainer
    b. Establish statistics to collect for performance dashboard
    c. Create screen sketches
    d. Develop basic graphics from screen sketches in Unity
    e. Code text and visuals for user guidance
        i. Decision tree
        ii. Button selections for next step or differential diagnosis (The user will be asked to choose what to do next or what diagnosis they want to make from a few options)
    f. Code detection that the user has completed evaluation and diagnosis steps correctly
        i. Detect if the user has moved the limb to the correct position as indicated (if applicable)
        ii. Detect is the user has chosen the next step incorrectly
        iii. Code statistics collection and store information in the database
    g. Code transition from module completion to performance dashboard (modified to display results for that specific quiz)
    h. Choose which sound clips to use and where to use them
7. Create user performance dashboard
    a. Establish which statistics will be used or not based on the module type
    b. Create screen sketches
    c. Develop basic graphics from screen sketches in Unity
    d. Code user-specific data retrieval from database
    e. Code necessary calculations to display the correct statistics
    f. Code statistics display
8. Set up database
    a. Design database architecture
        i. Decide on tables and columns
    b. Create database/set up server
    c. Populate with test data
9. Establish connection between VIRA and database
    a. Create Spring backend

10. Enhance graphics quality in Unity

3.4 | Possible Risks and Risk Management

There are a few different risks inherent to our project. First, because we were forced to switch to VR from AR, there is the risk that VIRA will not function as we initially envisioned. This means that we may be compromising on some implementation decisions to make VIRA work properly. Similarly, VIRA is designed for athletic training students and will thus require medical/anatomical accuracy that none of us are too familiar with. In order to mitigate the risk of our module not working as the client intends, we need to work closely with them.

3.5 | Proposed Project Milestones and Evaluation Criteria

Our milestones are the top-level items in our outline in Section 3.3. Our evaluation criteria are as follows:

- **Establish connection between VIRA and database:** VIRA can access the database for user profile selection and displaying the performance dashboard
- **Set up database:** Data can be stored and retrieved properly
- **Create user profile selection:** The user can select a profile under which to record their activities
- **Create user home screen:** The user can view and select one of three options: guided module, quiz module, and performance dashboard
- **Create module selection:** The user can select a module to work through
- **Develop models for the chosen limb(s)**: The user can view the limb(s) in skin, muscle, and tendon modes
- **Create quiz module:** The user can take a quiz and be evaluated on the tasks they perform without step-by-step guidance
- **Create user performance dashboard:** The user can view their progress on guided modules or statistics on their performance in recent quiz modules
- **Enhance graphics quality in Unity:** Graphics are smooth and realistic enough that the user can easily understand what is being displayed
- **Create guided special test module:** The user can navigate through a module and perform the tasks necessary to complete it, learning how to implement the special test they have chosen along the way

3.6 | Project Tracking Procedures

We are using GitLab as our code repository and to track progress during the project. High-level project progress will be tracked through GitLab via Milestones (See Section 3.5) and specific development tasks (See Section 3.3) will be represented as Issues, most of which will contribute to a certain Milestone. Unity also has version control available (Perforce, PlasticSCM), which we will decide upon as a team once development has begun.

3.7 | Expected Results and Validation

Our desired outcome is that we have at least one fully functional module that helps incorporate technology into an athletic training environment. We will know that VIRA works at a high-level through user acceptance testing. Once we begin the second semester, we plan to take the product to ISU's athletic training program at the end of each sprint to show our progress to the customers. We will use their feedback to ensure that we're working towards the correct solution. In the end, we will know that our solution is successful when it has been deemed medically accurate and effective by an athletic trainer.

4 | Project Timeline, Estimated Resources, and Challenges

4.1 | Project Timeline

Table 1 provides a summary of the Fall 2019 semester.

Table 1 – Summary of Completed Tasks

Month	Accomplishments
September	• Had initial team, advisor, and client meetings • Gathered requirements • Researched AR and VR platforms • Made hardware and software decisions • Made system design decision
October	• Learned more about working with Unity and the Oculus Quest • Established tasks • Refined system design
November	• Created screen sketches • Made decisions about the first special test • Established database architecture • Began making initial prototypes and tests in Unity
December	• Continued researching and prototyping • Completed end-of-semester assignments

Table 2 and *Table 3* show a schedule of the tasks listed in Section 3.3.

Table 2 – Schedule of Tasks (Fall Semester)

Task ID	Task Description	Estimated Time to Complete	Relevant Documentation
1A	Create user profile selection screen sketches	1 day	*Figure 3*, Page A-1
2A	Create user home screen sketches	1 day	*Figure 3*, Page A-1
3A	Create module selection screen sketches	1 day	*Figure 4*, Page A-2
4A	Develop models for the chosen limb(s)	6 weeks	
5B	Establish statistics to be collected from the guided special test modules for the performance dashboard	3 days	
5C	Create screen sketches for guided special test modules	1 day	*Figure 5*, Pages A-3, A-4

6B	Establish statistics to be collected from quiz module for performance dashboard	3 days	
7A	Establish which statistics will be used for the performance dashboard based on the module type	3 days	
8A	Design database architecture	3 days	*Figure 6*, Page A-

In *Table 3*, tasks of the same color can be completed in parallel.

Table 3 – Schedule of Tasks (Spring Semester)

Task ID	**Task Description**	**Estimated Time to Complete**	**Estimated Start Date**
8B	Create database/set up server	2 days	Jan. 13
2B	Develop basic graphics for user home from screen sketches	3 days	Jan. 13
5E	Code ability for user to interact with and move a limb in the guided module	2 weeks	Jan. 13
7B	Create screen sketches for user performance dashboard	1 day	Jan. 13
6C	Create screen sketches for quiz module	1 day	Jan. 13
7C	Develop basic graphics for user performance dashboard from screen sketches in Unity	2 weeks	Jan. 14
9A	Create a Spring backend to connect Unity to the database	4 days	Jan. 15 **(Milestone 9 complete)**
3B	Develop basic graphics for module selection from screen sketched	5 days	Jan. 16
1B	Develop basic graphics for user profile selection from screen sketches in Unity	3 days	Jan.16
5A	Establish narration/guided content of guided module with help from athletic trainer	2 days	Jan. 20
5D	Develop basic graphics for guided module from screen sketches in Unity	2 weeks	Jan. 20
6A	Establish narration for quiz module with help from athletic trainer	2 days	Jan. 20
8C	Populate database with test data	2 days	Jan. 21 **(Milestone 8 complete)**
5K	Choose which sound clips to use and when for guided module	1 week	Feb. 3
6H	Choose which sound clips to use and when for quiz module	1 week	Feb. 3

Task ID	Task Description	Estimated Time to Complete	Estimated Start Date
1C	Code population of user profile screen with users	1 week	Jan. 27
1D	Code transition from user selection screen to user home screen	2 days	Jan. 27 **(Milestone 1 complete)**
7D	Code retrieval of user-specific data from database	3 days	Jan. 27
3C	Code population of module options on module selection screen	1 week	Jan. 29
2C	Code transition of selection an option on the user home screen to its respective scene	2 days	Jan 29. **(Milestone 2 complete)**
3D	Code transition from module selection to starting the module	2 days	Jan. 31 **(Milestone 3 complete)**
5F	Code guided module dialogue and other visuals for user guidance	2 weeks	Feb. 3
5G	Code detection that user has followed guidance correctly and give feedback or redirect	2 weeks	Feb. 3
5J	Code transition from module completion to module selection screen	2 days	Feb. 3
6D	Develop basic graphics for quiz module from the screen sketches in Unity	1 week	Feb. 10
6E	Code quiz module prompts and other visuals based on user decisions	2 weeks	Feb. 17
6F	Code detection that user as complete evaluation and diagnosis steps correctly and collection of statistics based on their performance for quiz module	4 weeks	Mar. 2
6G	Come completion of quiz module to performance dashboard for that quiz	2 days	Mar. 30 **(Milestone 6 complete)**
7E	Code calculations needed to display the right statistics in the performance dashboard	1 week	Mar. 30
7F	Code displaying the statistics for the performance dashboard nicely	1 week	Apr. 6 **(Milestone 7 complete)**
4A	Create muscle model for the chosen limb	3 weeks	Jan. 27
4A	Create tendon model for the chosen limb	3 weeks	Mar. 30 **(Milestone 4 complete)**

Task ID	Task Description	Estimated Time to Complete	Estimated Start Date
5H	Code user muscle selection and responding identification (when user selects a muscle, its name appears)	6 weeks	Feb. 17
5I	Code user tendon selection and responding identification (when user selects a tendon, its name appears)	6 weeks	Apr. 20 (If time) **(Milestone 5 complete)**
10	Enhance graphics quality in Unity	2 weeks	Apr. 13 **(Milestone 10 complete)**

4.2 | Feasibility Assessment

We expect to complete two fully functional modules completed by the end of the school year: a guided special test module and a quiz module. This is a reasonable goal, and anything less would not be useful to the client. Our biggest challenge is the team's inexperience with VR development. We anticipate some challenges in finding suitable tutorials for certain things such as coding for the Oculus Quest, as well as difficulty working with the graphics. Because none of us have worked with VR before, we have added in a rather large buffer of time at the end of the spring semester to accommodate schedule changes if development goes slower than expected.

4.3 | Personal Effort Requirements

Table 4 below lists the tasks we have done and expect to do for this project. The numbers provided are very rough estimates and will vary from person to person and from week to week. Following are descriptions of the tasks.

Table 4 – Individual Tasks

Task	Estimated Time (Hours)
Gathering requirements	3 per sprint
Researching	12 hours
Acquiring hardware	4 (once)
Setting up development environment	0.5 (once)
Development	Variable
Testing	Variable
Writing design documents/presentations	5 per week

Gathering Requirements

The bulk of this task involves meeting with athletic trainers to discuss the next steps for the sprint. The time required for this will vary based on what we plan to accomplish that sprint and how familiar we are with the material required to do those things.

Researching

We will need to research on a continual basis; from researching which hardware to choose to researching the best means of adding a model in Unity, we will constantly be researching to ensure that we do not reinvent the wheel.

Acquiring Hardware

This will only need to be done once. We have successfully acquired our Oculus Quest and should not need any further hardware.

Setting up Development Environment

This is also a one-time task. Once we have our Oculus Quest, we can all make sure that our development environments are set up similarly. Once we all have Unity properly set up and are all connected to our GitLab repository, we should not have to do anything further for this task.

Development

This is, of course, the bulk of our project's work. Most of our time should be spent working on the module and communicating what each of us has accomplished. Since we have spent most of our time as a team sorting out which technologies we will be using and working on presentations and design documents, we are uncertain of how much time we will need to put into developing a VR application.

Testing

A significant amount of time will be spent on testing, including unit, system, regression, and acceptance testing. This is an important part of the development process and will help us follow continuous development practices. As we have yet to begin development, we are unsure of how much time will be spent testing.

Writing Design Documents and Presentations

This includes working on all project management overhead, design documents, and presentations that are required for this course. Based on our time spent thus far, we estimate that this will take roughly three to five hours per week.

4.4 | Other Resource Requirements

The project will require an Oculus Quest VR headset, controllers, and a carrying case. Additionally, we will require time to meet with athletic trainers to discuss requirements, allow them to test the program, and receive feedback from them. We will also be using a variety of software programs, namely: MakeHuman, Blender, and Unity Pro. The first two are free programs, while Unity Pro is a paid program for which we have obtained a free educational license.

4.5 | Financial Requirements

As this project is student-proposed, we do not have company funding like some other projects might. Thus, we are financially restricted to what the Department can provide. We used $540 of our budget on the Oculus Quest and a carrying case. Additionally, we have acquired licenses for the Pro version of the Unity engine, which is free for educational institutions.

5 | Testing and Implementation

5.1 | Interface Specifications

VIRA will be run solely on the Oculus Quest. A USB cable will be required to download the application from a computer onto the headset. The application will communicate with a remote Spring server, which will act as an interface between the application and the database. We will need to test the system as a whole to ensure that the wireless connection is adequate for retrieving data. Any other testing will not require any specific interfaces.

5.2 | Hardware and Software

We will not need any additional hardware or software for testing VIRA. The Unity development environment provides us with unit testing and debugging capabilities. It is useful for modeling how the module will run on the Quest as well as making it obvious when there are visual bugs. Testing on the Quest depends solely on the device, as we only have one. Acceptance testing will be performed by the end-users. We will also use Junit to test the Spring back-end layer in between the application and the database. We are considering implementing continuous integration (CI) and continuous development (CD) to limit the amount of time we spend resolving merge conflicts.

5.3 | Functional Testing

We plan to use Unity to perform unit and module testing. Our integration testing will be performed by an automated GitLab CI script that will perform a set of tests every time a team member pushes changes. System and acceptance testing will be completed by the team and the end-users.

5.4 | Non-Functional Testing

To test usability, we will analyze how first-time users navigate through VIRA to assess its intuitiveness. For performance testing, we will monitor running speed, response time, wireless connection, and battery consumption of repeated uses to ensure that VIRA runs efficiently on the Oculus Quest. Since VIRA will be developed specifically for the Oculus Quest, we do not expect any compatibility issues. Our main security concern regards the user accounts. We will ensure security in our database and will minimize network requests to reduce the chances that information is stolen.

5.5 | Process

As we have not yet begun development, we have not begun any sort of testing.

5.6 | Results

As we have not yet begun implementing our design, we have not begun any sort of testing.

6 | Closing Material

6.1 | Conclusion

So far, we have worked toward designing our project, gathering and refining our requirements, and deciding upon our technologies. We have also made basic design decisions regarding the user interfaces and the system architecture. We began learning how to use the Unity environment and work with the Oculus Quest. We have experimented with using MakeHuman and Blender to create realistic humanoid models and are currently in the initial prototyping stage. Our ultimate goals are to:

1. develop at least two complete VIRA module,
2. design a modular product to allow for easy extensibility, and
3. learn about VR development.

We have decided upon the Oculus Quest as our hardware because it is the best option for our budget. We have chosen Unity as our development environment because it is an industry standard for developing 3D applications such as ours.

6.2 | References

[1] "IEEE Announces Virtual Reality (VR) and Augmented Reality (AR) Standards Projects in Advance of Participation at Augmented World Expo," *IEEE Standards Association*, 09-May-2017. [Online]. Available: https://standards.ieee.org/news/2017/ieee_p2408.html. [Accessed: 31-Oct-2019].

[2] "Quest Virtual Reality Check (VRC) Guidelines," *Oculus Developer Center*. [Online]. Available: https://developer.oculus.com/distribute/latest/concepts/publish-quest-req/. [Accessed: 31-Oct-2019].

[3] "2048.5 - Standard for Virtual Reality and Augmented Reality: Environment Safety," *IEEE Standards Association*, 21-Mar-2019. [Online]. Available: https://standards.ieee.org/project/2048_5.html. [Accessed: 04-Dec-2019].

[4] "2048.6 - Standard for Virtual Reality and Augmented Reality: Immersive User Interface," *IEEE Standards Association*, 21-Mar-2019. [Online]. Available: https://standards.ieee.org/project/2048_6.html. [Accessed: 04-Dec-2019].

Appendices

Appendix A | Screen Sketches

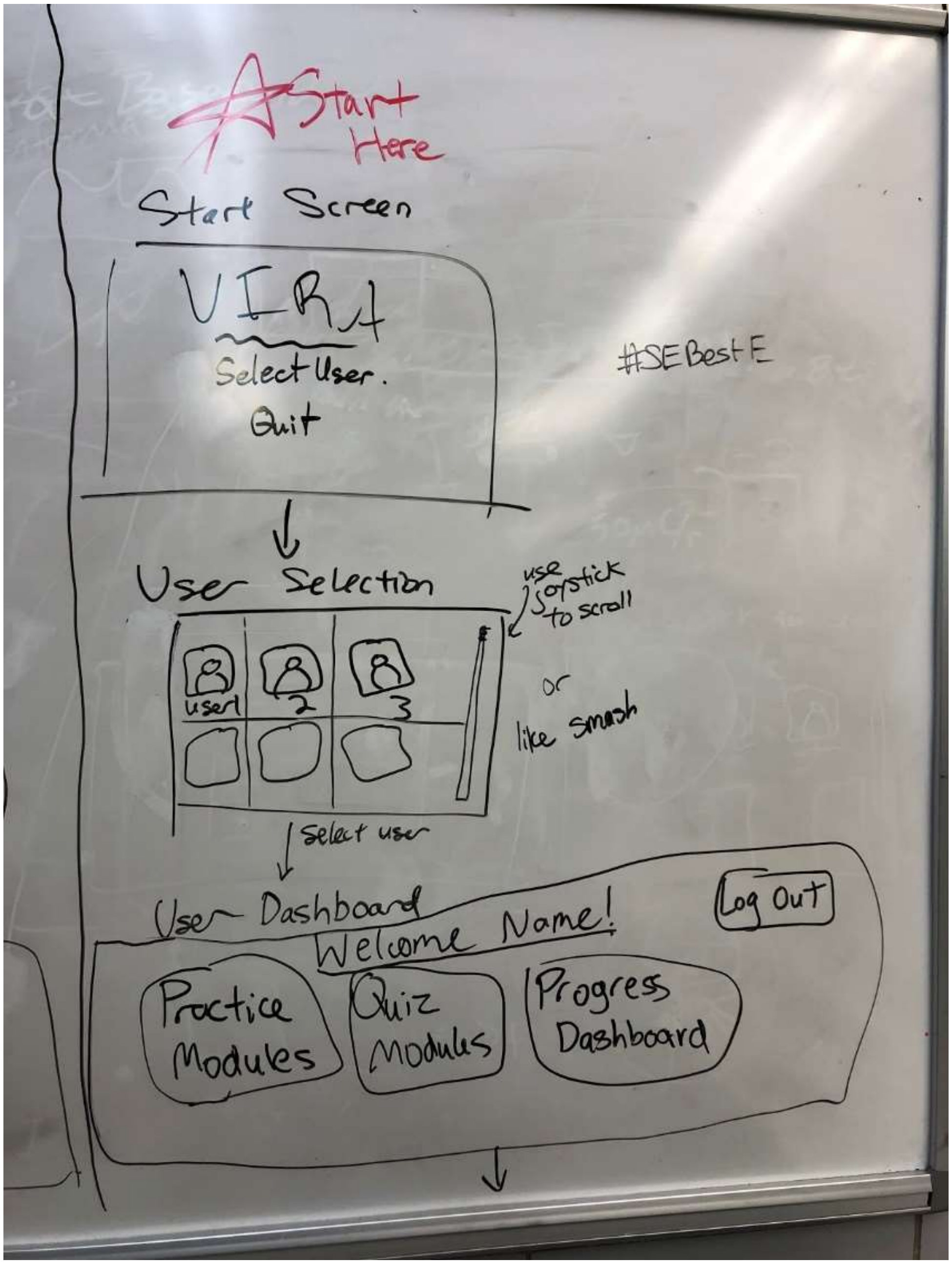


Figure 3 – Start Screen, User Selection, User Dashboard

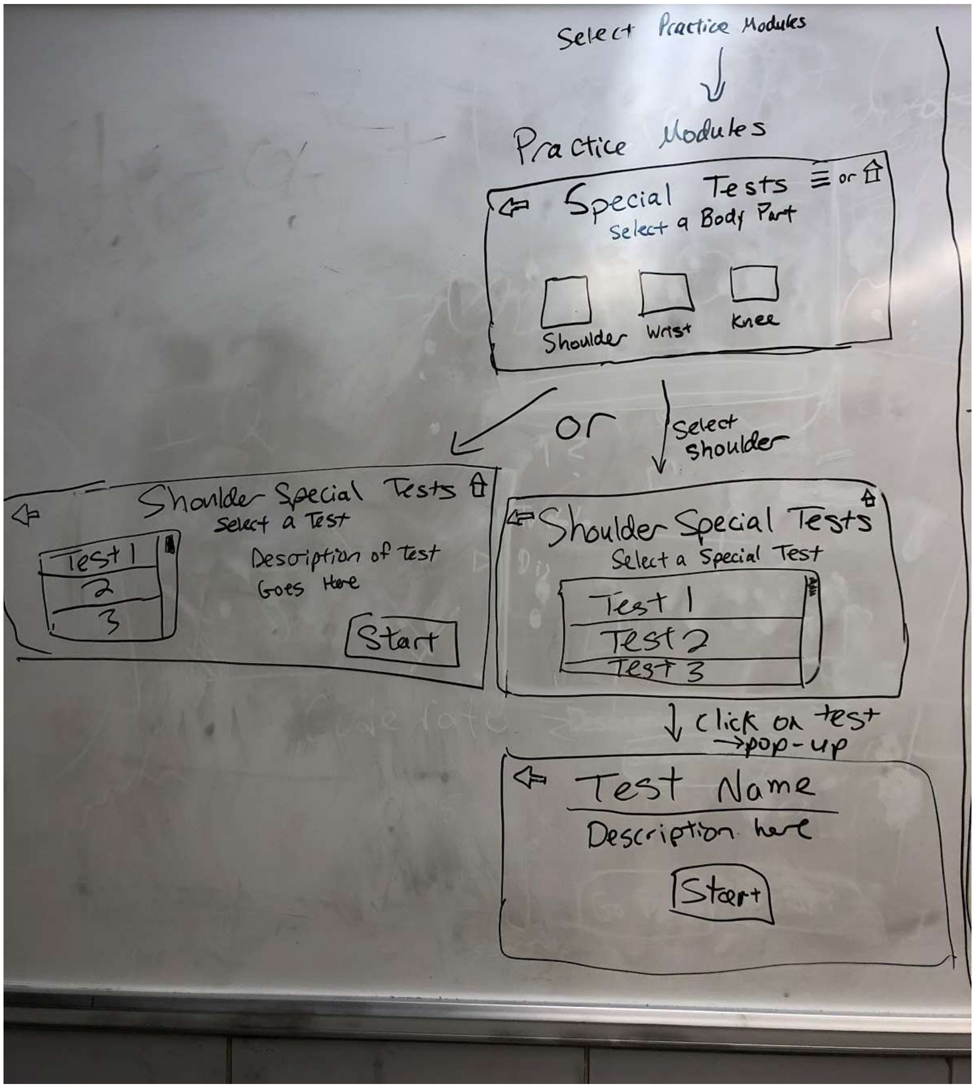
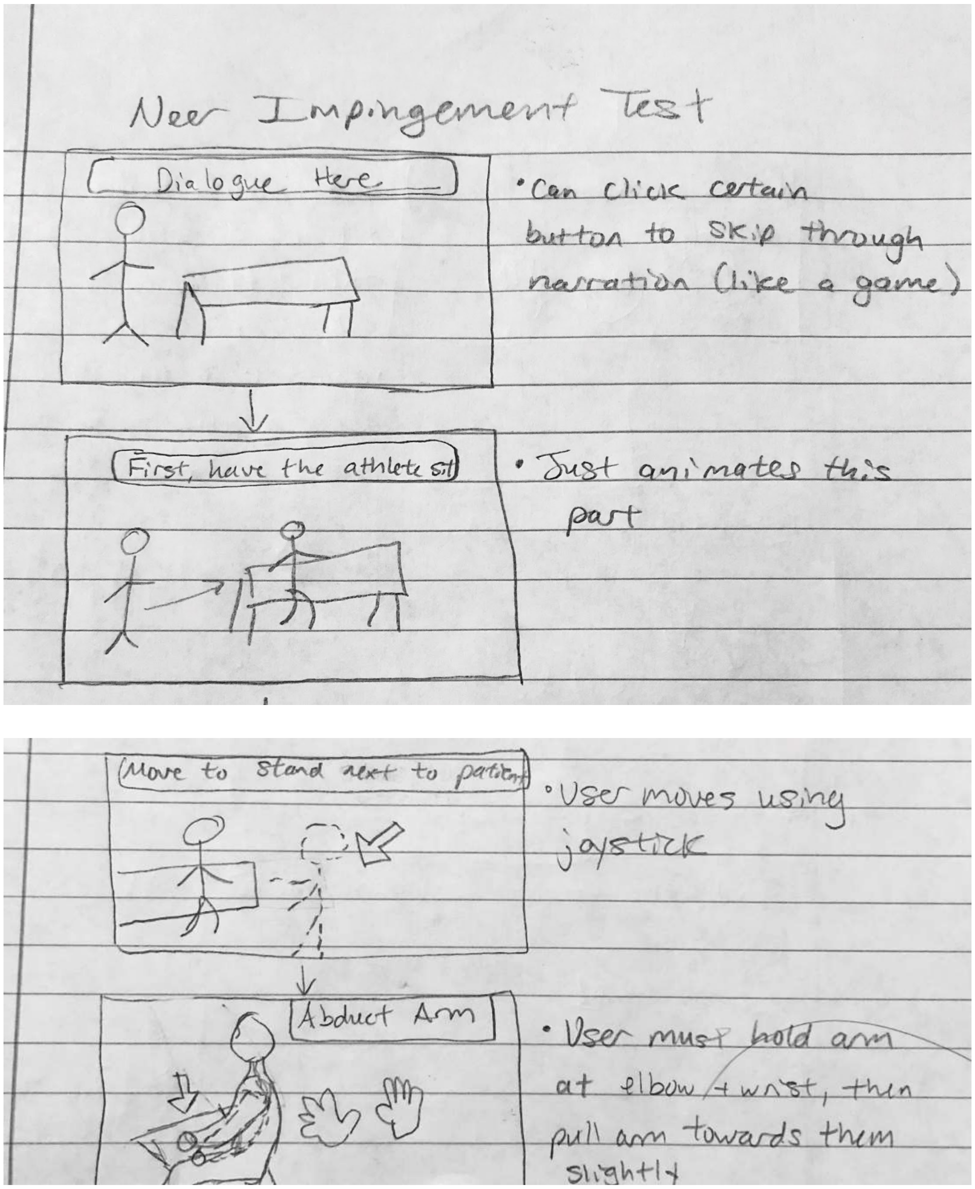


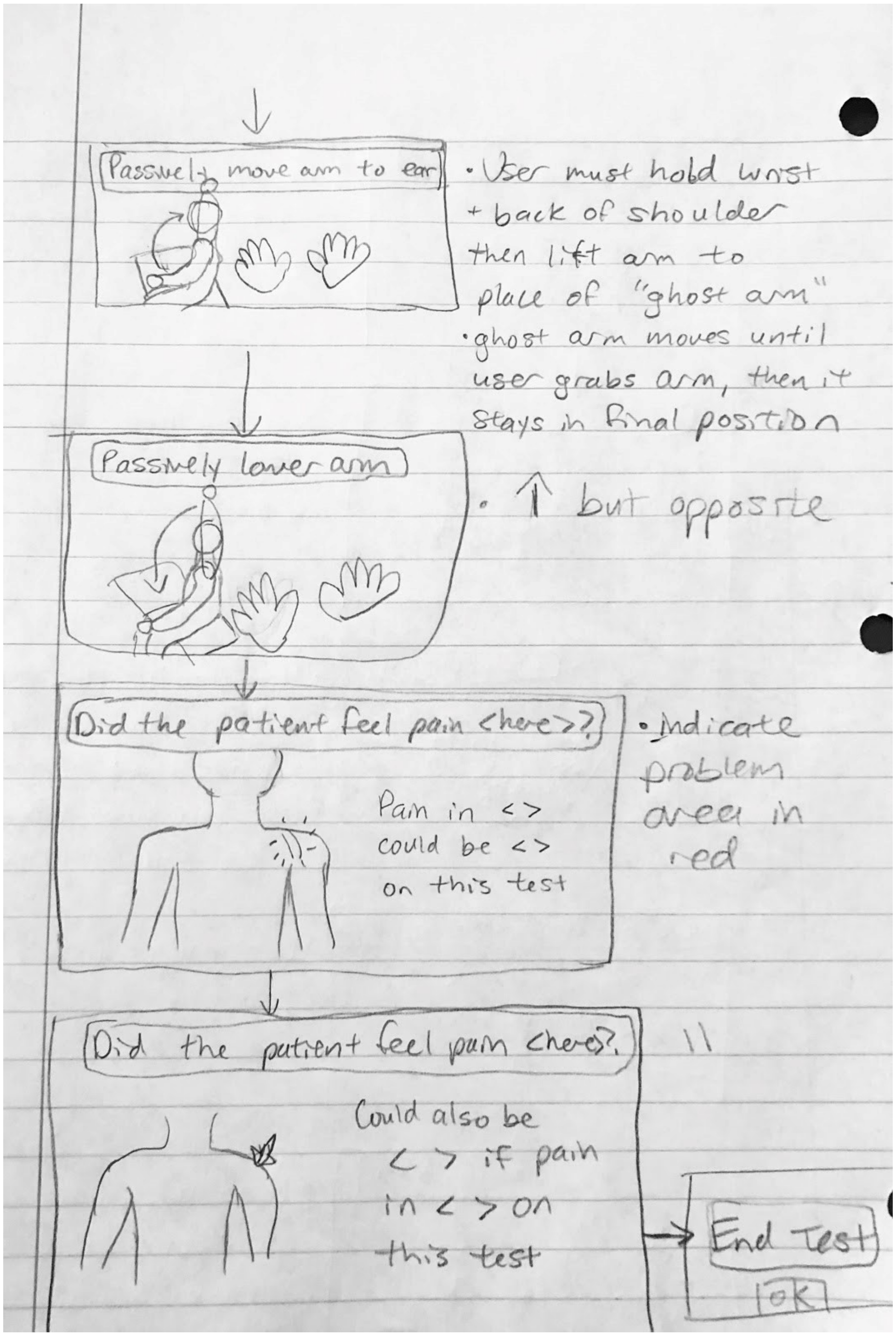
Figure 4 – Module Selection



Neer Impingement Test

- Can click certain button to skip through narration (like a game)
- Just animates this part
- User moves using joystick
- User must hold arm at elbow + wrist, then pull arm towards them slightly

Figure 5 – Practice Module


Passwely move arm to ear
Passively lower arm
Did the patient feel pain <here>?
Did the patient feel pain <here>?
End Test
OK

• User must hold wrist + back of shoulder then lift arm to place of "ghost arm"

• ghost arm moves until user grabs arm, then it stays in final position

• ↑ but opposite

Pain in <> could be <> on this test

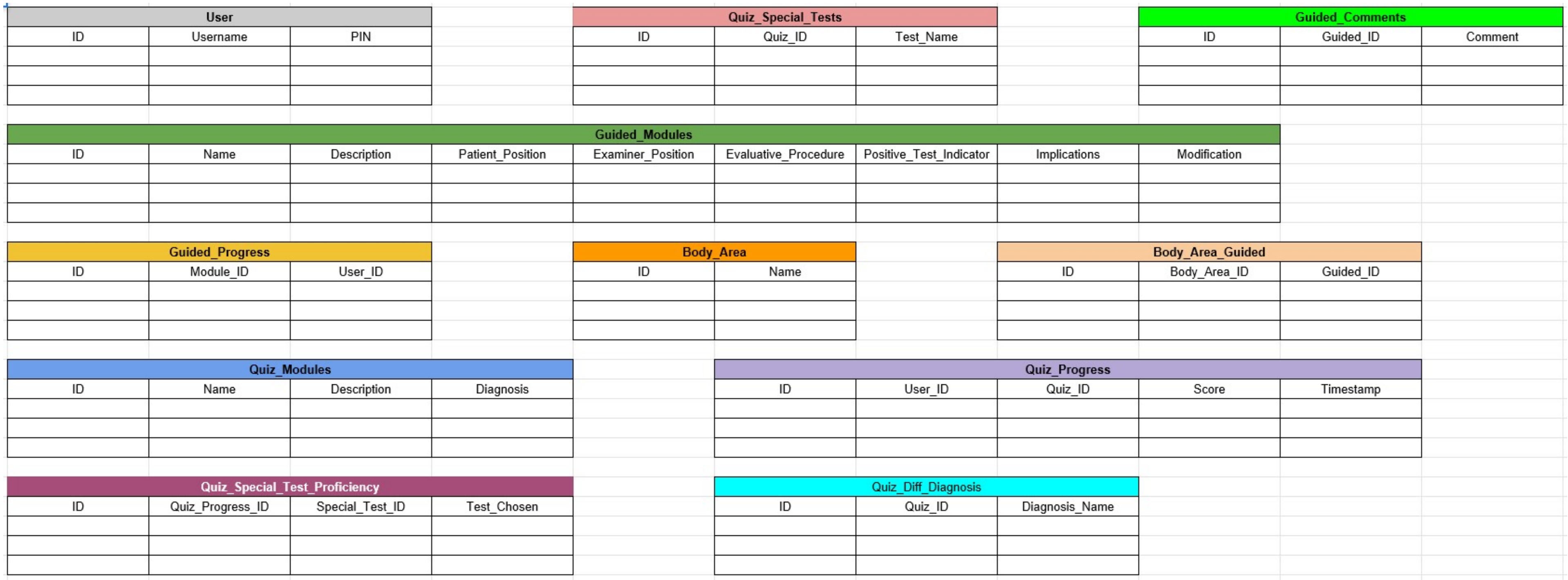
• Indicate problem area in red

Could also be <> if pain in <> on this test

\\

Figure 5 – Practice Module (cont.)

Appendix B | Database Architecture



User		
ID	Username	PIN

Quiz_Special_Tests		
ID	Quiz_ID	Test_Name

Guided_Comments		
ID	Guided_ID	Comment

Guided_Modules								
ID	Name	Description	Patient_Position	Examiner_Position	Evaluative_Procedure	Positive_Test_Indicator	Implications	Modification

Guided_Progress		
ID	Module_ID	User_ID

Body_Area	
ID	Name

Body_Area_Guided		
ID	Body_Area_ID	Guided_ID

Quiz_Modules			
ID	Name	Description	Diagnosis

Quiz_Progress				
ID	User_ID	Quiz_ID	Score	Timestamp

Quiz_Special_Test_Proficiency			
ID	Quiz_Progress_ID	Special_Test_ID	Test_Chosen

Quiz_Diff_Diagnosis		
ID	Quiz_ID	Diagnosis_Name

Figure 6 – Database Tables