Module Guide for Software Engineering

Team #22, TeleHealth Insights
Mitchell Weingust
Parisha Nizam
Promish Kandel
Jasmine Sun-Hu

April 3, 2025

1 Revision History

Date	Version	Member	Notes
1/13/2025	1.0	Mitchell Weingust	Added 10 - Clinician Dashboard Interfaces
1/13/2025	1.1	Promish Kandel	Added 5 - Module Hierarchy
1/13/2025	1.2	Parisha Nizam	Added 10 - Home Page
1/14/2025	1.3	Parisha Nizam	Added Module Decomposition
1/14/2025	1.4	Jasmine Sun-Hu	Added 7 - Module Decomposition, Anticipated and Unlikely Changes, and User Interfaces
1/14/2025	1.5	Mitchell Weingust	Added 12 - Timeline
1/14/2025	1.6	Mitchell Weingust	Added 10 - Clinician Dashboard FSM
1/14/2025	1.7	Parisha Nizam	Added 8 - Traceability Matrix
1/14/2025	1.8	Mitchell Weingust	Added 7 - Module Decomposition
1/14/2025	1.9	Promish Kandel	Added 9 - Use Hierarchy Between Modules
1/14/2025	1.10	Promish Kandel	Added 7 - Module Decomposition
1/15/2025	1.11	Everyone	FSM Machine and Review

Date	Version	Member	Notes
03/23/2025	2.0	Promish Kandel	Implemented TA Feedback: Added AC3 to account for exact hardware changes
03/23/2025	2.1	Promish Kandel	Implemented TA Feedback: Removed input/output devices as these can change, as shown in AC3
03/23/2025	2.2	Promish Kandel	Implemented TA Feedback: Updated the wording for UC1 to be more descriptive rather than just "core structure"
03/23/2025	2.3	Promish Kandel	Implemented TA Feedback: Removed bilingual as it doesn't matter in the context of unlikely changes
03/23/2025	2.4	Promish Kandel	Implemented TA Feedback: AppController was fully removed and replaced with Landing Page GUI which is now a behaviour hinding module.
03/23/2025	2.5	Promish Kandel	Implemented TA Feedback: Added a logical module showcase our hardware hiding due to the browser. Also made note that this is an external module, not something we implemented. Showcased in section 7.1 Hardware hiding modules
03/23/2025	2.6	Promish Kandel	Implemented TA Feedback: Spelling was fixed and wording was made more consistent
03/23/2025	2.7	Promish Kandel	Implemented TA Feedback: Section 5 and traceability matrix are hyperlinked. I also fixed the hyperlink to point to the respective module decompostion rather than section 5

Date	Version	Member	Notes
03/23/2025	2.8	Promish Kandel	Implemented TA Feedback: Module Hierarchy diagram was added as a pdf to allow for zooming in and to make it more clear. This diagram now also contains Landing Page GUI and Logical Module as an external module
03/23/2025	2.9	Promish Kandel	Implemented TA Feedback: Update the secrets for 7.2.7 Question Banks Module
03/23/2025	2.10	Promish Kandel	Removed Logging, RealTimeFeedback and Report Generation modules based on changing design decisions.
04/03/2025	2.11	Mitchell Weingust	Reviewed document and fixed formatting

2 Reference Material

This section records information for easy reference.

2.1 Abbreviations and Acronyms

symbol	description
AC	Anticipated Change
DAG	Directed Acyclic Graph
M	Module
MG	Module Guide
OS	Operating System
R	Requirement
SC	Scientific Computing
SRS	Software Requirements Specification
Software Engineering	Explanation of program name
UC	Unlikely Change
FSM	Finite State Machine

Contents

1 Revision History **i**

2 Reference Material **iv**
2.1 Abbreviations and Acronyms iv

3 Introduction **1**

4 Anticipated and Unlikely Changes **2**
4.1 Anticipated Changes 2
4.2 Unlikely Changes 2

5 Module Hierarchy **3**

6 Connection Between Requirements and Design **3**

7 Module Decomposition **3**
7.1 Hardware Hiding Modules 4
7.1.1 Logical module 4
7.2 Behaviour-Hiding Module 5
7.2.1 Clinician GUI (M1) 5
7.2.2 Parent GUI (M7.2.2) 5
7.2.3 Landing Page GUI (M7.2.3) 5
7.2.4 Authentication Module (M7.2.4) 5
7.2.5 Result Storage Module (M7.2.5) 6
7.2.6 Media Processing Module (M7.2.6) 6
7.2.7 Question Bank Module (M7.2.7) 6
7.2.8 Video Processing Module (M7.2.8) 7
7.2.9 Audio Processing Module (M7.2.9) 7
7.2.10 English Question Bank Module (M7.2.10) 7
7.2.11 Mandarin Question Bank Module (M7.2.11) 7
7.2.12 Matching Question Bank Module (M7.2.12) 8
7.2.13 Repetition Question Bank Module (M7.2.13) 8
7.3 Software Decision Module 8
7.3.1 API Gateway Module (M7.3.1) 8

8 Traceability Matrix **8**

9 Use Hierarchy Between Modules **10**

10 User Interfaces **12**

11 Design of Communication Protocols **23**

12 Timeline **24**

List of Tables

1 Module Hierarchy 4
2 Trace Between Requirements and Modules 9
3 Trace Between Anticipated Changes and Modules 10

List of Figures

1 Use hierarchy among modules 11
2 Clinician Dashboard 12
3 Add Client 12
4 Patient Overview 13
5 Patient Assessment Results Analysis (1) 13
6 Patient Assessment Results Analysis (2) 14
7 Bias Review 14
8 Flag Bias 15
9 Login or Create an Account 15
10 Create an Account 16
11 Login in to account 16
12 Parent HomePage 16
13 Sketch of Assessment Selection Page 17
14 Sketch of Parent Checklist Page 17
15 Sketch of Consent Popup 18
16 Sketch of Video Audio and Mic Test Page 18
17 Sketch of Example Question Page 19
18 Sketch of Assessment Completion Page 19
19 FSM - TeleHealth Insights System 20
20 FSM - Clinician Dashboard 21
21 FSM - Assessment Dashboard 22

3 Introduction

Decomposing a system into modules is a commonly accepted approach to developing software. A module is a work assignment for a programmer or programming team (Parnas et al., 1984). We advocate a decomposition based on the principle of information hiding (Parnas, 1972). This principle supports design for change, because the "secrets" that each module hides represent likely future changes. Design for change is valuable in SC, where modifications are frequent, especially during initial development as the solution space is explored.

Our design follows the rules layed out by Parnas et al. (1984), as follows:

- System details that are likely to change independently should be the secrets of separate modules.
- Each data structure is implemented in only one module.
- Any other program that requires information stored in a module's data structures must obtain it by calling access programs belonging to that module.

After completing the first stage of the design, the Software Requirements Specification (SRS), the Module Guide (MG) is developed (Parnas et al., 1984). The MG specifies the modular structure of the system and is intended to allow both designers and maintainers to easily identify the parts of the software. The potential readers of this document are as follows:

- New project members: This document can be a guide for a new project member to easily understand the overall structure and quickly find the relevant modules they are searching for.
- Maintainers: The hierarchical structure of the module guide improves the maintainers' understanding when they need to make changes to the system. It is important for a maintainer to update the relevant sections of the document after changes have been made.
- Designers: Once the module guide has been written, it can be used to check for consistency, feasibility, and flexibility. Designers can verify the system in various ways, such as consistency among modules, feasibility of the decomposition, and flexibility of the design.

The rest of the document is organized as follows. Section 4 lists the anticipated and unlikely changes of the software requirements. Section 5 summarizes the module decomposition that was constructed according to the likely changes. Section 6 specifies the connections between the software requirements and the modules. Section 7 gives a detailed description of the modules. Section 8 includes two traceability matrices. One checks the completeness of the design against the requirements provided in the SRS. The other shows the relation between anticipated changes and the modules. Section 9 describes the use relation between modules.

4 Anticipated and Unlikely Changes

This section lists possible changes to the system. According to the likeliness of the change, the possible changes are classified into two categories. Anticipated changes are listed in Section 4.1, and unlikely changes are listed in Section 4.2.

4.1 Anticipated Changes

Anticipated changes are the source of the information that is to be hidden inside the modules. Ideally, changing one of the anticipated changes will only require changing the one module that hides the associated decision. The approach adapted here is called design for change.

AC1: The supported languages for the assessments (e.g., adding languages like Spanish, French, etc.).

AC2: The types of assessments supported (e.g., addition of new assessment types like sentence completion, story-telling, etc.).

AC3: The exact hardware that is going to be used (e.g., the type of microphone, type of camera for video recording).

4.2 Unlikely Changes

The module design should be as general as possible. However, a general system is more complex. Sometimes this complexity is not necessary. Fixing some design decisions at the system architecture stage can simplify the software design. If these decision should later need to be changed, then many parts of the design will potentially need to be modified. Hence, it is not intended that these decisions will be changed.

UC1: The structure of our question bank system and the way questions and their associated audio assets are formatted and organized in our database is designed for long-term stability and is unlikely to change.

UC2: The primary purpose of the system, which is to assist parents in administering speech assessments for children.

5 Module Hierarchy

This section provides an overview of the module design. Modules are summarized in a hierarchy decomposed by secrets in Table 1. The modules listed below, which are leaves in the hierarchy tree, are the modules that will actually be implemented.

M7.2.1: Clinician GUI Module

M7.2.2: Parent GUI Module

M7.2.3: Landing Page GUI

M7.3.1: API Gateway Module

M7.2.4: Authentication Module

M7.2.5: Result Storage Module

M7.2.6: Media Processing Module

M7.2.7: Question Bank Module

M7.2.8: Video Processing Module

M7.2.9: Audio Processing Module

M7.2.10: English Question Bank Module

M7.2.11: Mandarin Question Bank Module

M7.2.12: Matching Question Bank Module

M7.2.13: Repetition Question Bank Module

6 Connection Between Requirements and Design

The design of the system is intended to satisfy the requirements developed in the SRS. In this stage, the system is decomposed into modules. The connection between requirements and modules is listed in Table 2.

7 Module Decomposition

Modules are decomposed according to the principle of "information hiding" proposed by Parnas et al. (1984). The *Secrets* field in a module decomposition is a brief statement of the design decision hidden by the module. The *Services* field specifies *what* the module will do without documenting *how* to do it. For each module, a suggestion for the implementing software is given under the *Implemented By* title. If the entry is *OS*, this means that the module is provided by the operating system or by standard programming language libraries. *Software Engineering* means the module will be implemented by the Software Engineering software.

Only the leaf modules in the hierarchy have to be implemented. If a dash (–) is shown, this means that the module is not a leaf and will not have to be implemented.

Level 1	Level 2
Hardware-Hiding	Logical module
Behaviour-Hiding	Clinician GUI
	Parent GUI
	Landing Page GUI
	Authentication Module
	Result Storage Module
	Real-Time Feedback Module
	Report Generation Module
	Media Processing Module
	Video Processing Module
	Audio Processing Module
	Logging Module
	Question Bank Module
	Mandarin Question Bank
	English Question Bank
	Repetition Question Bank Module
	Matching Question Bank Module
Software Decision	API Gateway

Table 1: Module Hierarchy

7.1 Hardware Hiding Modules

7.1.1 Logical module

Secrets: This logical module is handled by the browser, which is outside the scope of our project. The browser internally manages and isolates both audio and microphone feeds from direct access.

Services: Although not part of our implementation, the browser ultimately supplies audio and microphone data to the Parent GUI (M7.2.2) when a user is taking a test.

Implemented By: Standard web browser

Type of Module: External

7.2 Behaviour-Hiding Module

7.2.1 Clinician GUI (M1)

Secrets: The interactive and visual components that allow Clinicians to interact with the system, through the Landing Page GUI (M7.2.3), to access patient data and information, and make informed decisions.

Services: To show application functionality to clinicians, accepting user inputs (choosing assessments to review, flagging bias questions) and displaying outputs (assessment summaries).

Implemented By: ClinicianFrontEnd

Type of Module: Library

7.2.2 Parent GUI (M7.2.2)

Secrets: The interactive and visual components that allow Parents to interact with the system, through the Landing Page GUI (M7.2.3), to set up and engage in the assessment with their child.

Services: To show application functionality to parents, accepting user inputs (selecting answers to questions, completing set up) and displaying outputs (question visuals, button selections).

Implemented By: ParentFrontEnd

Type of Module: Library

7.2.3 Landing Page GUI (M7.2.3)

Secrets: The interactions between the GUIs (M7.2.1, M7.2.2) and the API Gateway (M7.3.1), acting as a means to interface with the software modules.

Services: Enables the user to pass information from the GUIs to the backend services.

Implemented By: AppController

Type of Module: Library

7.2.4 Authentication Module (M7.2.4)

Secrets: The data structures and algorithms used to securely store, validate, and manage user credentials.

Services: Provides user registration, login, and session management services. Ensures authentication for all system users (parents, clinicians, and admins) to maintain system security.

Implemented By: AuthenticationService

Type of Module: Library, Abstract Data Type

7.2.5 Result Storage Module (M7.2.5)

Secrets: The schema and mechanisms used to store, index, and retrieve assessment results and metadata efficiently.

Services: Manages the storage and retrieval of processed media flags, assessment results, and associated metadata. Ensures data security and organization to support reporting and feedback functionalities.

Implemented By: ResultStorageService

Type of Module: Record, Abstract Object

7.2.6 Media Processing Module (M7.2.6)

Secrets: The design and implementation of how media (video and audio) is processed in the system.

Services: Provides high-level functionality for media processing by delegating tasks to its submodules: Video Processing Module and Audio Processing Module. Acts as an abstraction layer for handling media data.

Implemented By: Media-processing-service

Type of Module: Abstract Object

7.2.7 Question Bank Module (M7.2.7)

Secrets: Maintains all question bank submodules and the internal logic for routing each request to the appropriate submodule.

Services: Acts as a facade to provide unified access to all question banks. This module handles requests for retrieving, adding, updating, or delegating questions to appropriate submodules.

Implemented By: QuestionBankService

Type of Module: Abstract Object

7.2.8 Video Processing Module (M7.2.8)

Secrets: The methods and algorithms used to process video data, including frame extraction, format handling, and metadata processing.

Services: Handles all video-related data processing tasks, such as analyzing video frames, ensuring quality, and extracting relevant details. This module communicates with the Media Processing Module.

Implemented By: Media-processing-service

Type of Module: Abstract Object

7.2.9 Audio Processing Module (M7.2.9)

Secrets: The methods and algorithms used to process audio data, such as format conversions, noise filtering, and speech analysis.

Services: Handles all audio-related data processing tasks, including speech detection, sound quality analysis, and extracting key audio features. This module communicates with the Media Processing Module.

Implemented By: Media-processing-service

Type of Module: Abstract Object

7.2.10 English Question Bank Module (M7.2.10)

Secrets: The format for storing, tagging and/or indexing English questions

Services: Converts the input data into the data structure used by the input parameters module.

Implemented By: EnglishQuestionManager

Type of Module: Abstract Data Type

7.2.11 Mandarin Question Bank Module (M7.2.11)

Secrets: The format and structure of the input data.

Services: Converts the input data into the data structure used by the input parameters module.

Implemented By: MandarinQuestionManager

Type of Module: Abstract Data Type

7.2.12 Matching Question Bank Module (M7.2.12)

Secrets: The format and structure of the input data.

Services: Converts the input data into the data structure used by the input parameters module.

Implemented By: MatchingQuestionService

Type of Module: Library

7.2.13 Repetition Question Bank Module (M7.2.13)

Secrets: The format and structure of the input data.

Services: Converts the input data into the data structure used by the input parameters module.

Implemented By: MatchingQuestionService

Type of Module: Library

7.3 Software Decision Module

7.3.1 API Gateway Module (M7.3.1)

Secrets: The interactions between the Landing Page GUI (M7.2.3) and the inter-dependencies of all other software modules, including inherited modules (M7.2.5, M7.2.6, M7.2.7, M7.2.9, M7.2.10, M7.2.11, M7.2.12, M7.2.13).

Services: Enables the user to access the system and interact with its components, consisting of the Patient, Client, and Admin views.

Implemented By: APIGateway

Type of Module: Library

8 Traceability Matrix

This section shows two traceability matrices: between the modules and the requirements and between the modules and the anticipated changes.

Req.	Modules
FR-A1	M7.2.1, M7.2.2, M7.2.3, M7.2.4, M7.2.7
FR-A2	M7.2.2, M7.2.3, M7.2.4, M7.2.7
FR-A3	M7.2.1, M7.2.3, M7.2.4, M7.2.7
FR-A4	M7.2.4, M7.2.7
FR-A5	M7.2.4, M7.2.7
FR-SS1	M7.2.2, M7.2.3
FR-SS2	M7.2.2, M7.2.3
FR-SS3	M7.2.2, M7.2.3
FR-SS4	M7.2.2, M7.2.3
FR-SS5	M7.2.2, M7.2.3, M7.2.8
FR-AI1	M7.2.2, M7.2.3, M7.2.6, M7.2.8, M7.2.11, M7.2.12, M7.2.13
FR-AI2	M7.2.2, M7.2.3, M7.3.1, M7.2.6, M7.2.8, M7.2.11, M7.2.12, M7.2.13
FR-AI3	M7.2.2, M7.2.3, M7.3.1, M7.2.6, M7.2.8, M7.2.11, M7.2.12, M7.2.13
FR-AI4	M7.2.2, M7.2.3, M7.3.1, M7.2.6, M7.2.8, M7.2.11, M7.2.12, M7.2.13
FR-AI5	M7.2.2, M7.2.3, M7.3.1, M7.2.6, M7.2.8, M7.2.11, M7.2.12, M7.2.13
FR-AI6	M7.2.2, M7.2.3, M7.3.1, M7.2.5, M7.2.6, M7.2.8, M7.2.11, M7.2.12, M7.2.13
FR-AI7	M7.2.2, M7.2.3, M7.3.1, M7.2.6, M7.2.8, M7.2.11, M7.2.12, M7.2.13
FR-DSC1	M7.3.1, M7.2.5, M7.2.7
FR-DSC2	M7.3.1, M7.2.5, M7.2.6, M7.2.9, M7.2.11, M7.2.12
FR-DSC3	M7.3.1, M7.2.4
FR-DSC4	M7.3.1, M7.2.4
FR-DSC5	M7.3.1, M7.2.7, M7.2.9, M7.2.10
FR-VADA1	M7.3.1, M7.2.6, M7.2.11, M7.2.12
FR-VADA2	M7.3.1, M7.2.6, M7.2.7, M7.2.11, M7.2.12
FR-VADA3	M7.3.1, M7.2.6, M7.2.7, M7.2.11, M7.2.12
FR-DPD1	M7.3.1, M7.2.5, M7.2.7, M7.2.9, M7.2.10
FR-DPD2	M7.3.1, M7.2.5, M7.2.7, M7.2.9, M7.2.10
FR-DPD3	M7.2.1, M7.3.1, M7.2.5, M7.2.7, M7.2.9, M7.2.10
FR-DPD4	M7.2.1, M7.3.1, M7.2.5, M7.2.7, M7.2.9, M7.2.10

Table 2: Trace Between Requirements and Modules

AC	Modules
AC1	M7.2.8, M7.2.13
AC2	M7.2.8, M7.2.12, M7.2.13
AC3	M7.2.2

Table 3: Trace Between Anticipated Changes and Modules

9 Use Hierarchy Between Modules

In this section, the uses hierarchy between modules is provided. Parnas (1978) said of two programs A and B that A *uses* B if correct execution of B may be necessary for A to complete the task described in its specification. That is, A *uses* B if there exist situations in which the correct functioning of A depends upon the availability of a correct implementation of B. Figure 1 illustrates the use relation between the modules. It can be seen that the graph is a directed acyclic graph (DAG). Each level of the hierarchy offers a testable and usable subset of the system, and modules in the higher level of the hierarchy are essentially simpler because they use modules from the lower levels.

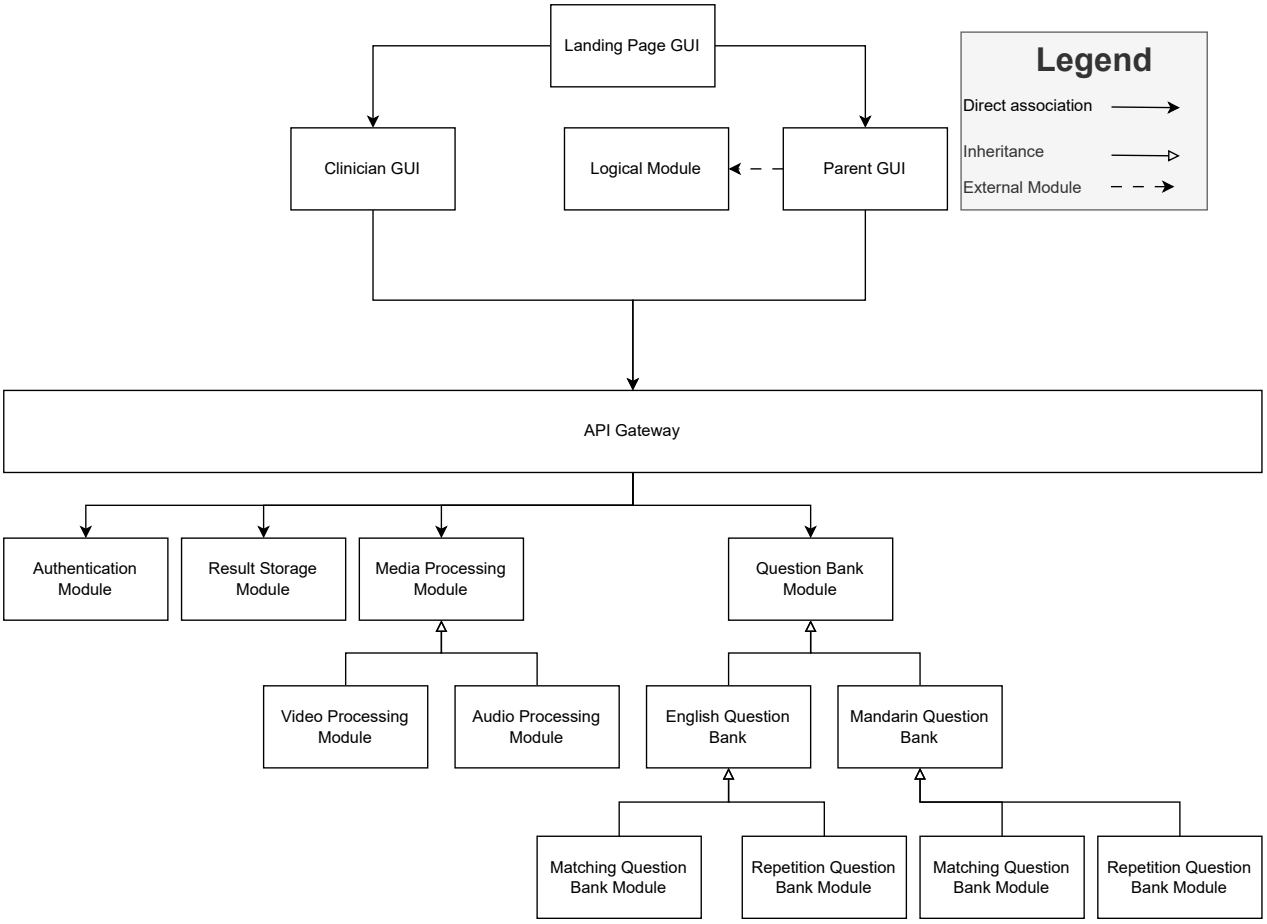

Figure 1: Use hierarchy among modules

10 User Interfaces

The interface below depicts the initial interface a clinician would see upon logging into their account in the system.

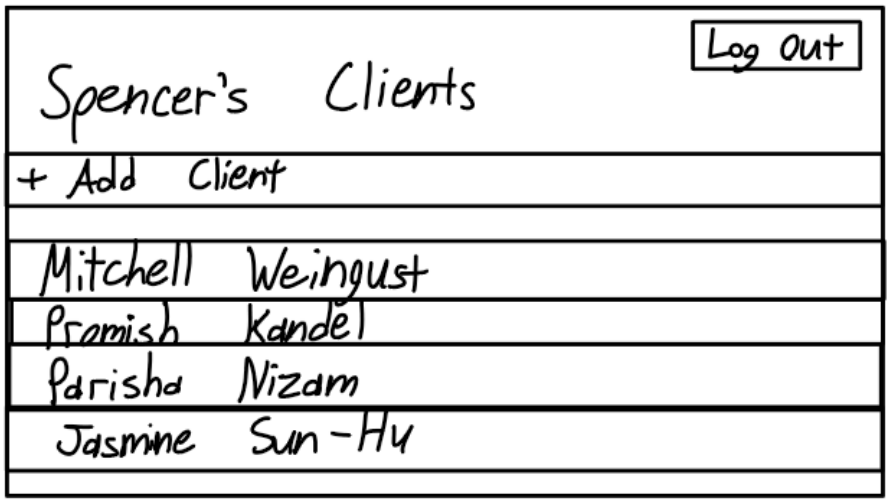

Figure 2: Clinician Dashboard

The interface below depicts the interface a clinician would see upon selecting the Add Client button on the previous Clinician Dashboard screen.

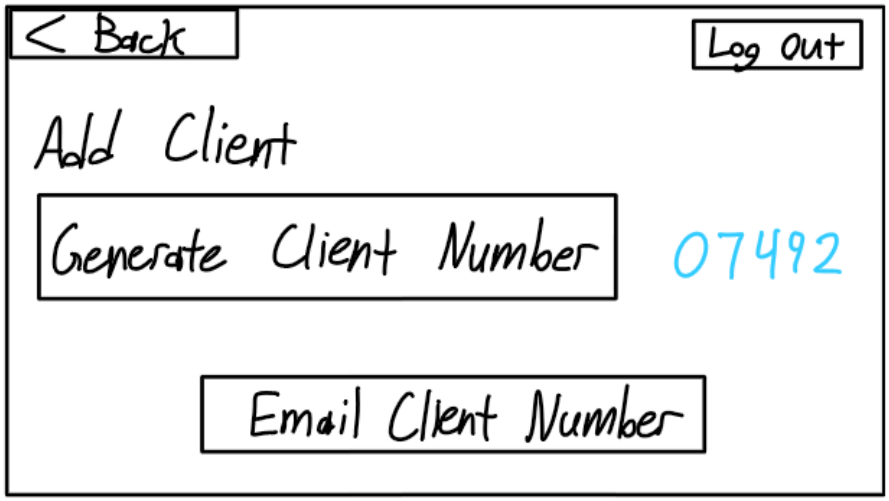

Figure 3: Add Client

The interface below depicts the patient overview, which can be reached from the Clinician Dashboard by selecting a name from the client list.

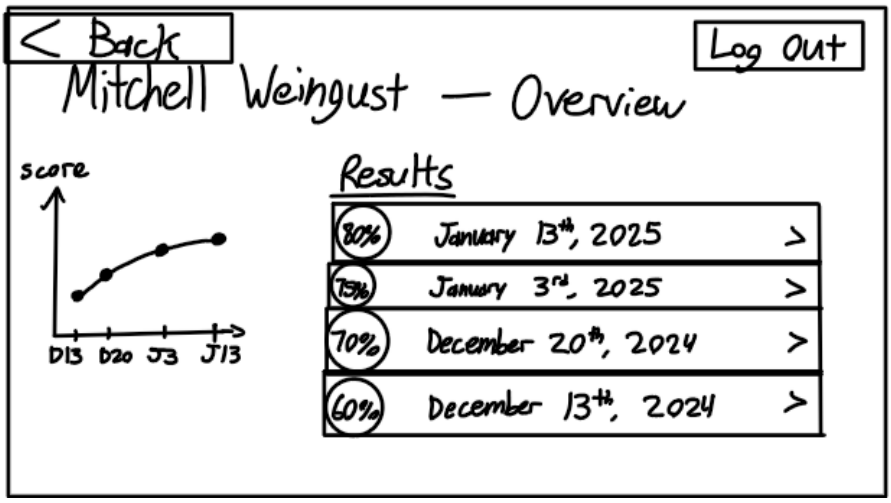

Figure 4: Patient Overview

The interface below depicts the patient assessment results analysis, which can be reached from the Patient Overview by selecting an assessment date from the list of assessments.

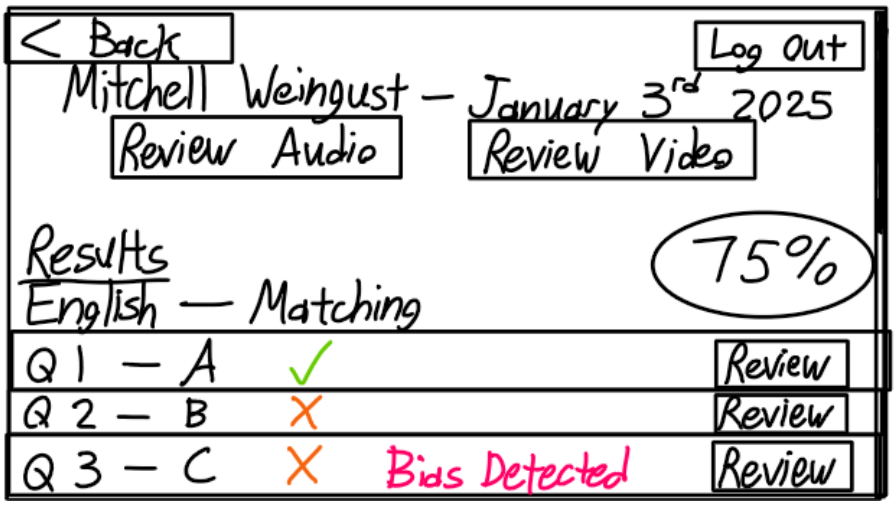

Figure 5: Patient Assessment Results Analysis (1)

The interface below depicts a continuation of the patient assessment results analysis, which can be reached from the previous figure, by scrolling the scrollbar on the right edge of the screen.

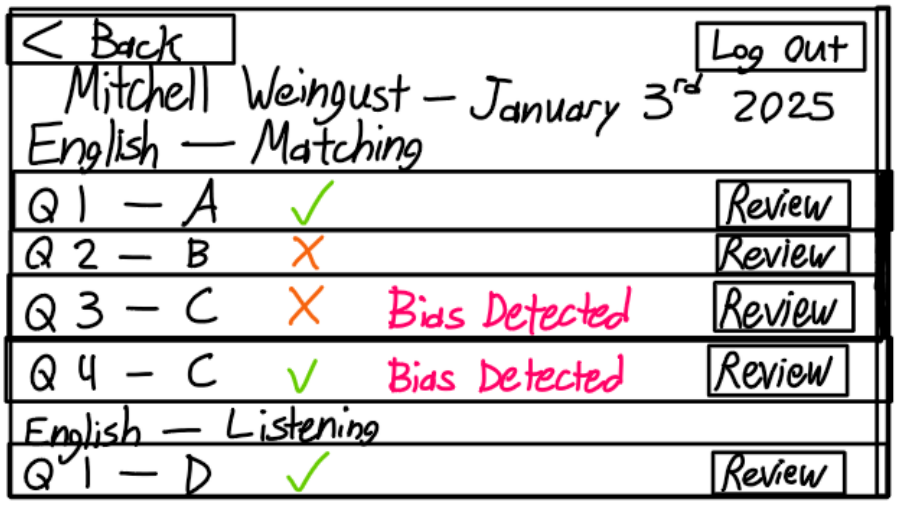

Figure 6: Patient Assessment Results Analysis (2)

The interface below depicts the bias review, which can be reached from the Patient Assessment Results Analysis by selecting Review on any of the questions on an assessment.

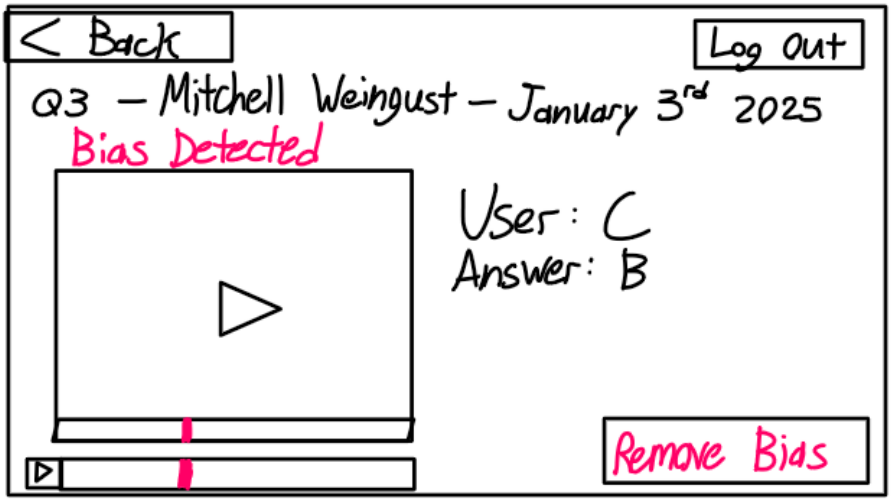

Figure 7: Bias Review

The interface below depicts a question review page, where no bias has been detected. The ability to Flag Bias is present in the bottom right corner, to give the Clinician the ability to manually reflect bias in a question.

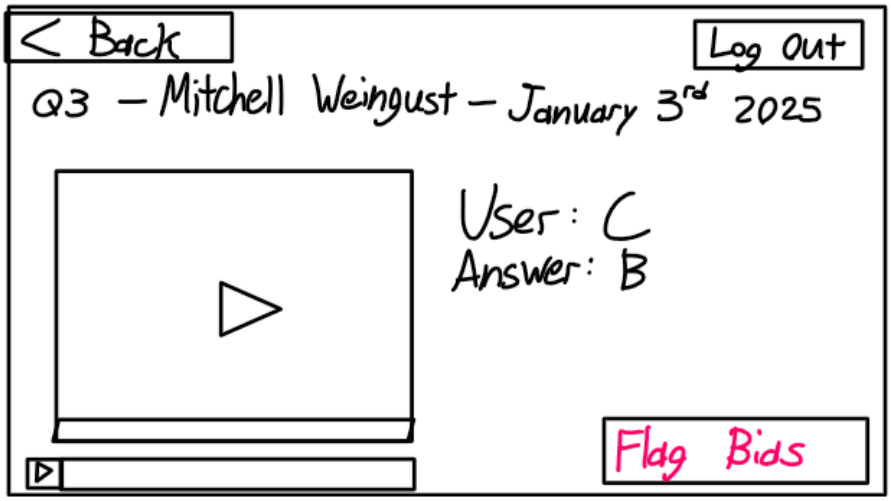

Figure 8: Flag Bias

The interfaces below depicts the interface allowing a user who enters the application to either login to the platform if they have an existing account, or create a new account for new users.

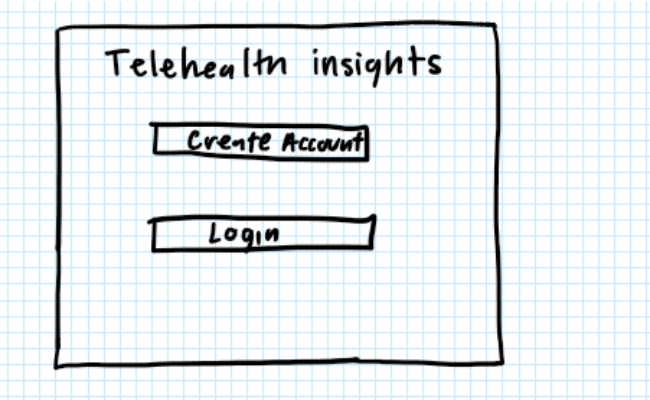

Figure 9: Login or Create an Account

The interfaces below depicts the flow of selecting which account type to create. If a parent account is chosen, they are able to create a username and password and enter client number to complete the account creation. A clinician account information with be created and provided to the clinician by the admin.

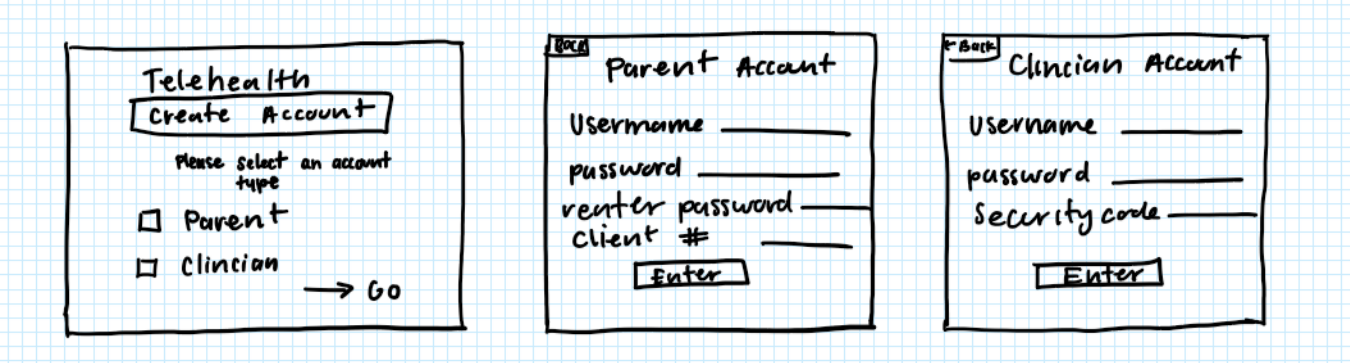

Figure 10: Create an Account

The interface below depicts the login page overview, where a user can login to the application if they already have an existing account.

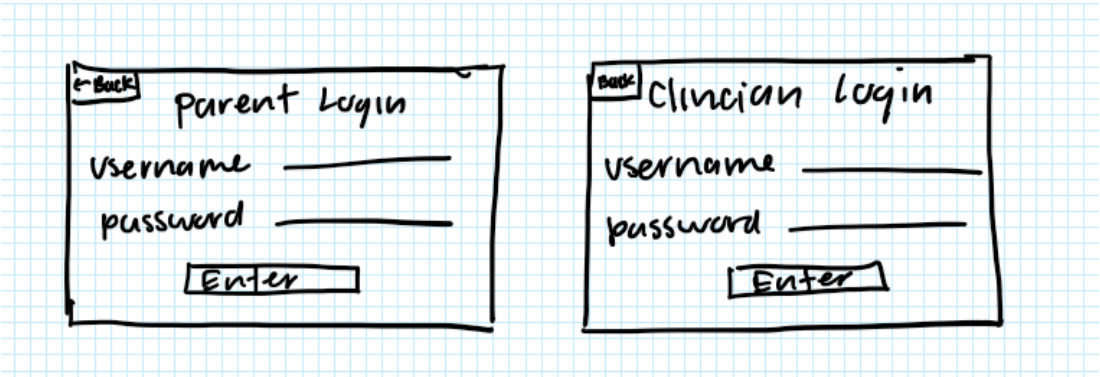

Figure 11: Login in to account

The interface below depicts the home page for the parent to enter the assesement platform. The home page provides options to learn how to use the assessment platform or start the assessment.

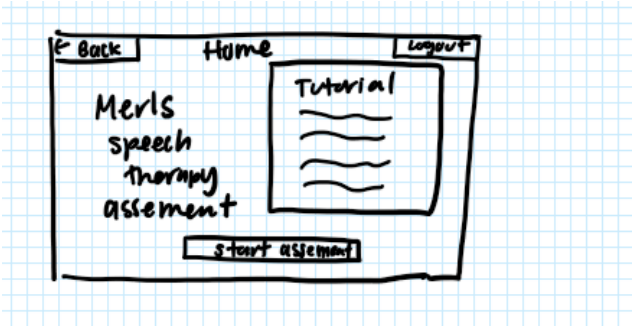

Figure 12: Parent HomePage

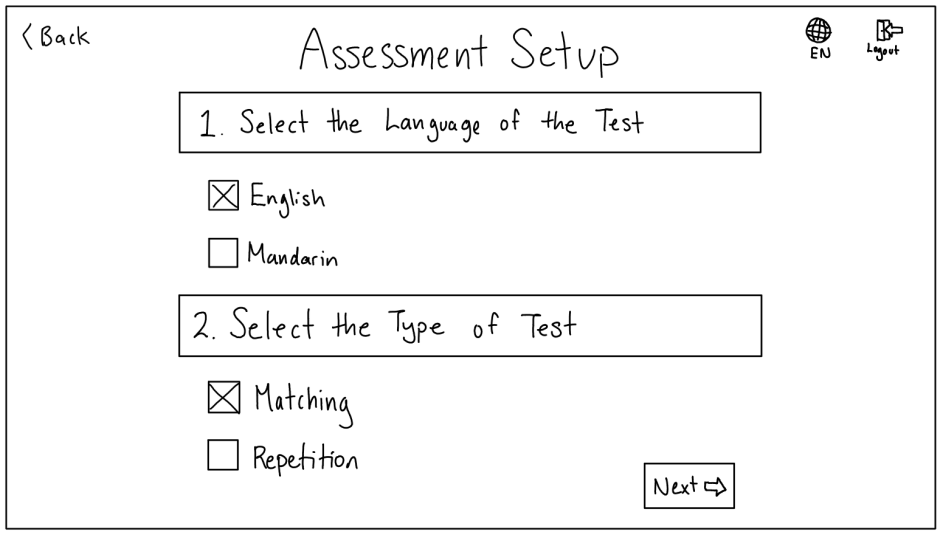

Figure 13: Sketch of Assessment Selection Page

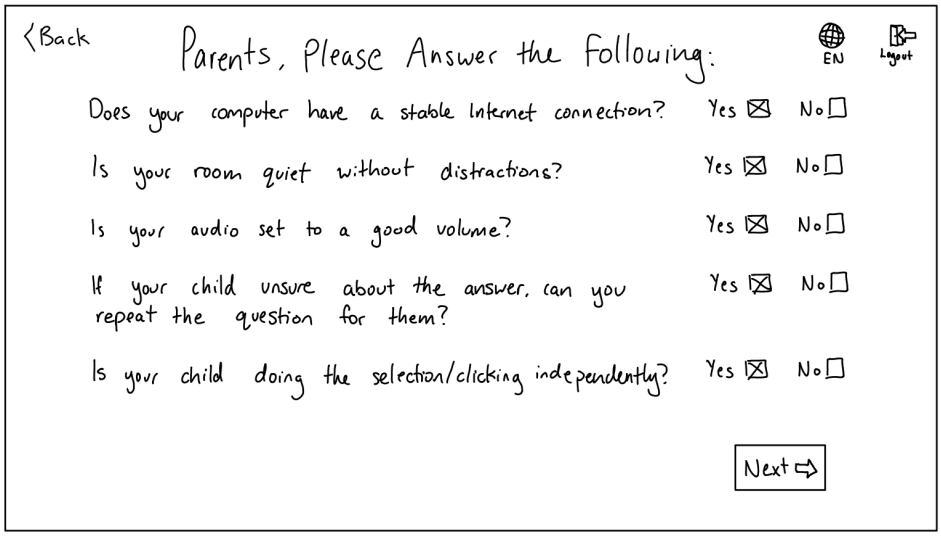

Figure 14: Sketch of Parent Checklist Page

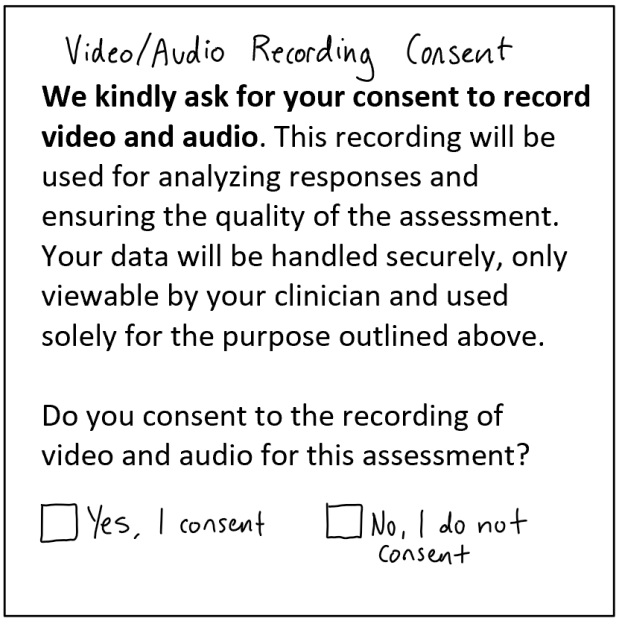

Figure 15: Sketch of Consent Popup

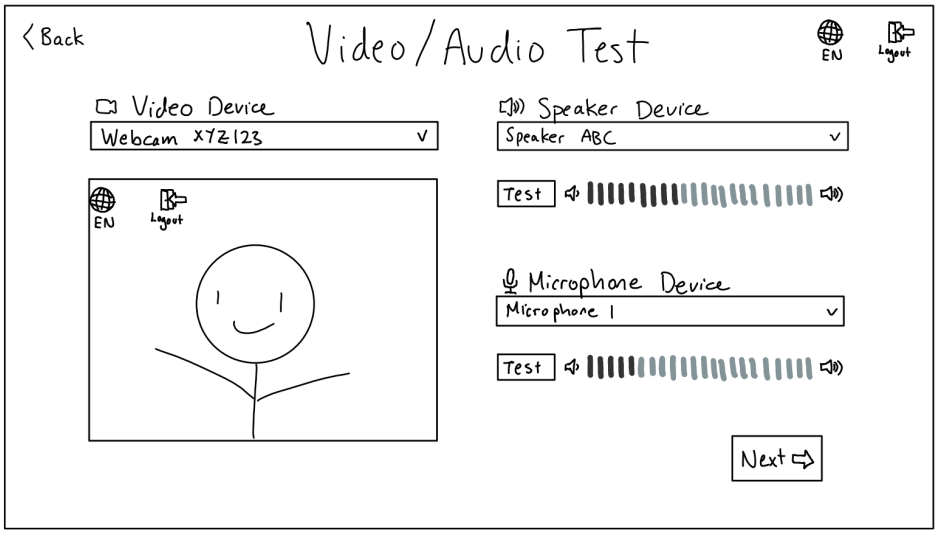

Figure 16: Sketch of Video Audio and Mic Test Page

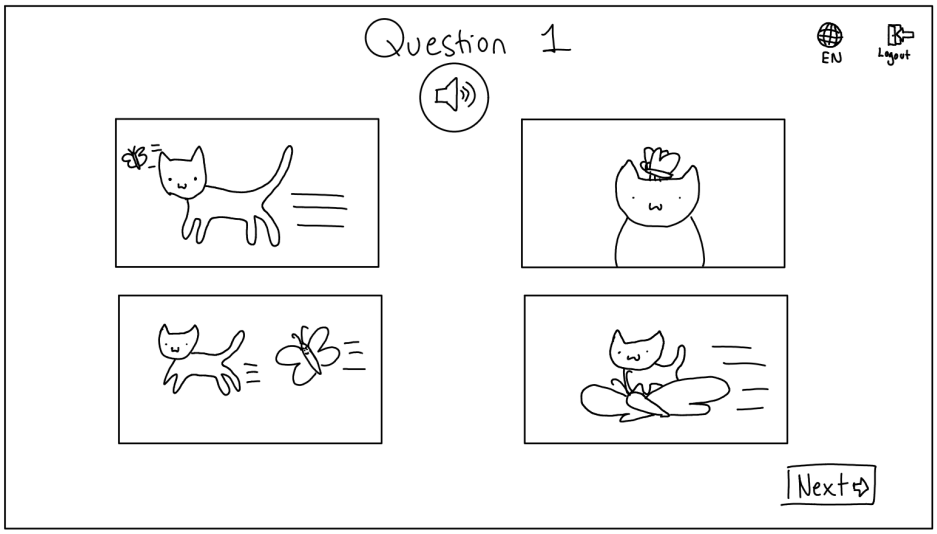

Figure 17: Sketch of Example Question Page

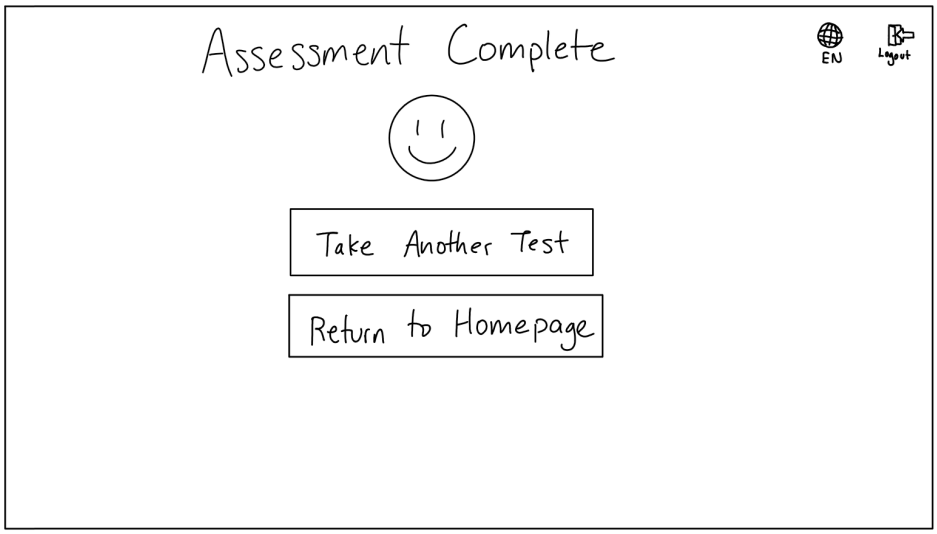

Figure 18: Sketch of Assessment Completion Page

The below finite state machine depicts how the overall system can be interacted with, as well as which actions lead to changes in states in the system. Included in this Finite State Machine are Clinician Dashboard and Assessment, which are further expanded in Figure 20 and Figure 21.

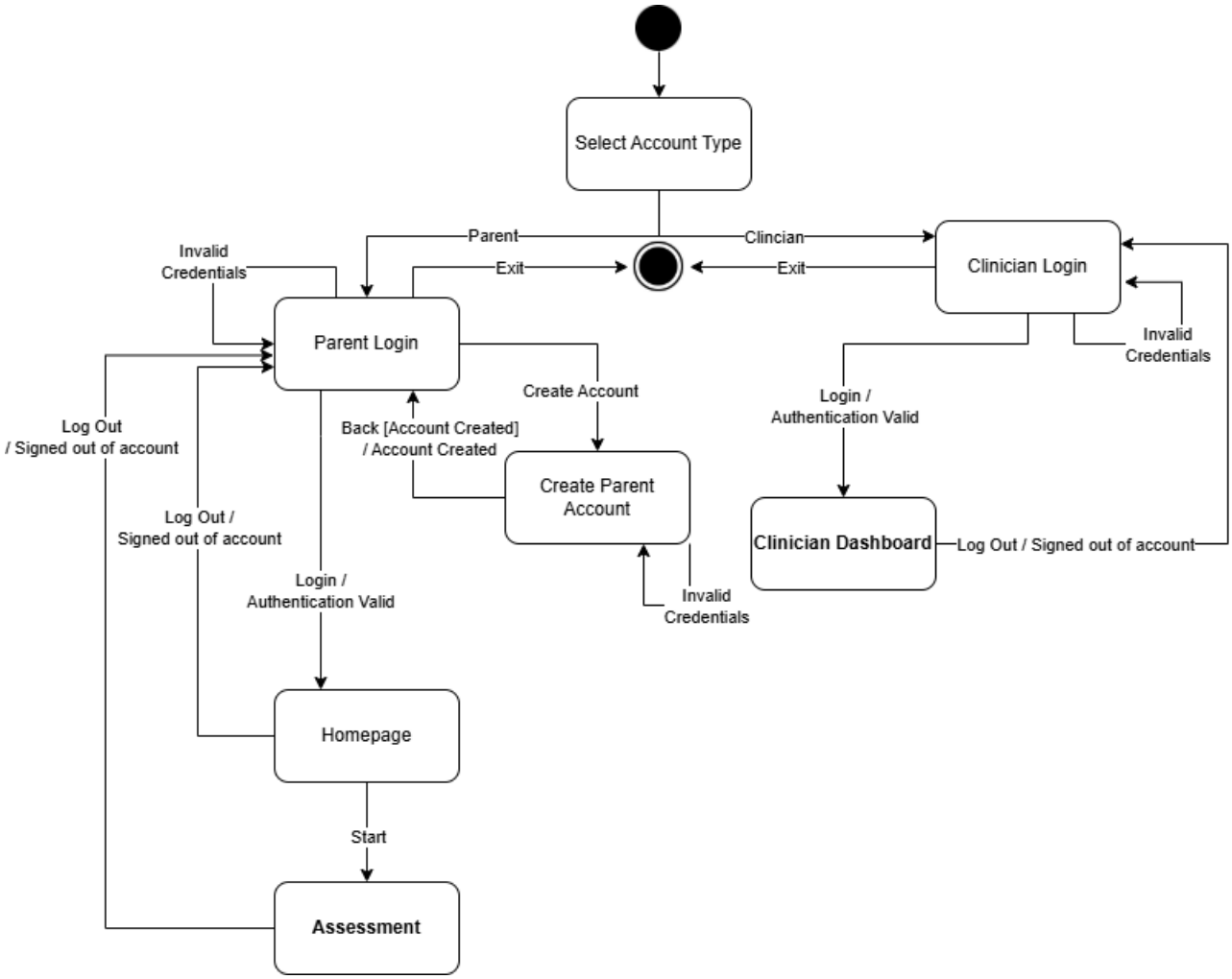

Figure 19: FSM - TeleHealth Insights System

The below finite state machine depicts how the clinician can interface with the dashboard, as well as which interactions lead to changes in states in the system.

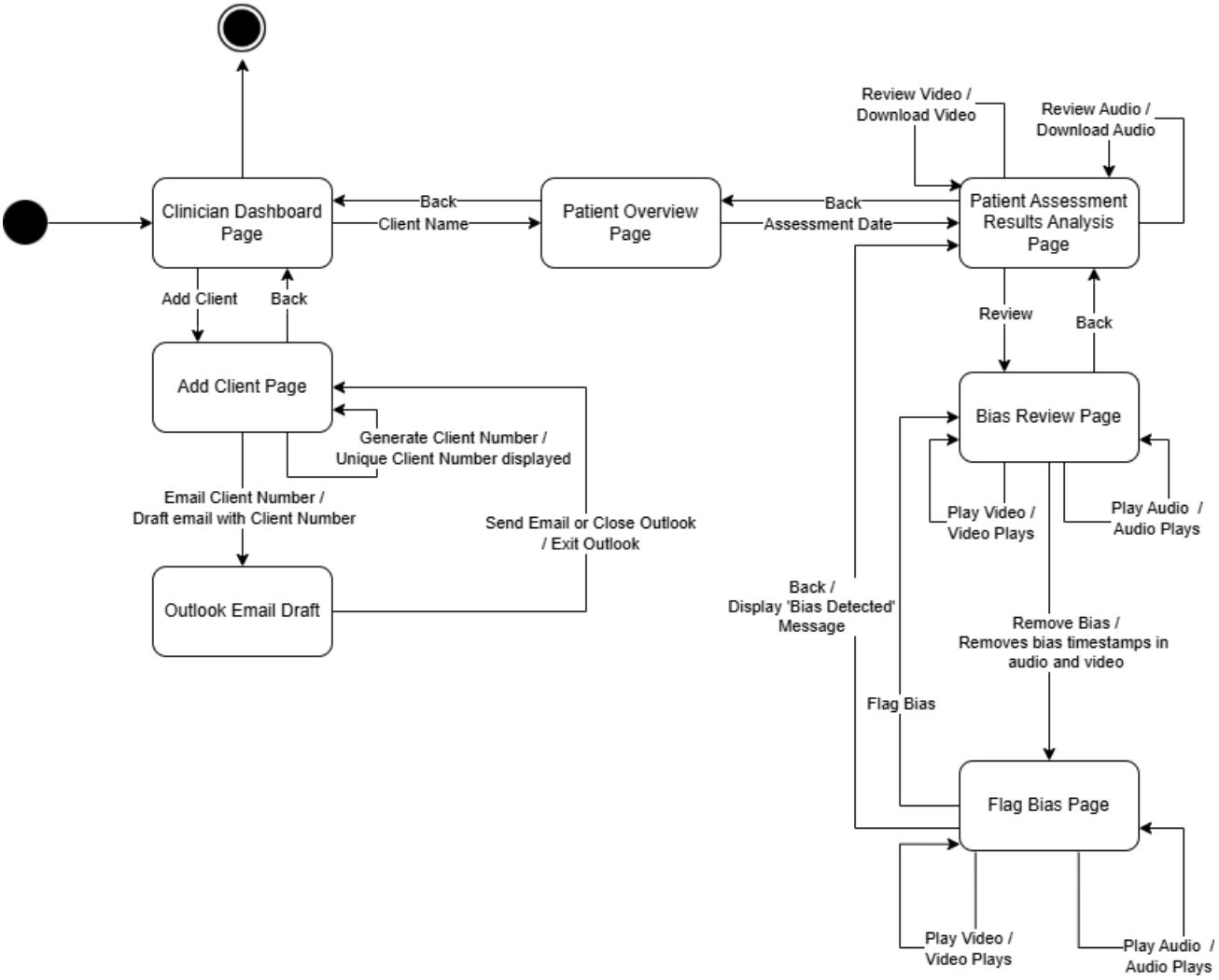

Figure 20: FSM - Clinician Dashboard

The below finite state machine depicts how the parent and child can interface with the assessment, as well as which interactions lead to changes in states in the system.

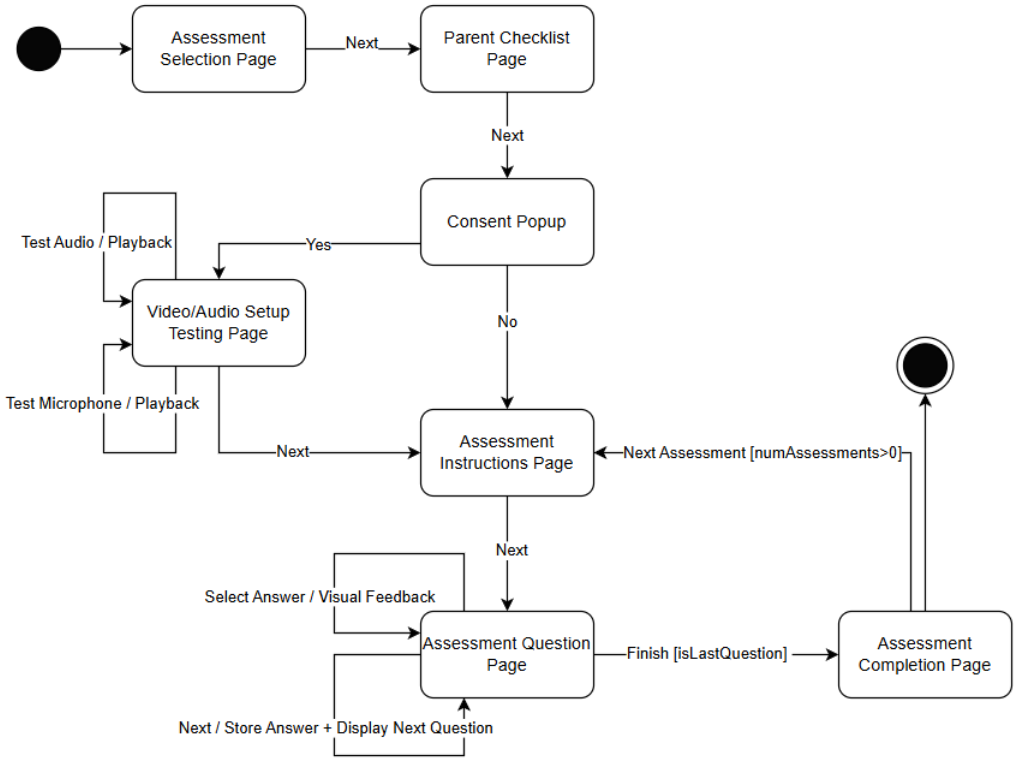

Figure 21: FSM - Assessment Dashboard

11 Design of Communication Protocols

N/A

12 Timeline

Milestone	Module/Pages	Objective	Mitchell	Parisha	Promish	Jasmine	Date
Controllers	API Gateway				X		1/19/25
Assessment	Question Bank Module		X	X			1/19/25
Assessment	English Question Bank Module		X	X			1/19/25
Assessment	Matching Question Bank Module		X	X			1/19/25
Assessment	Repetition Question Bank Module		X	X			1/19/25
Assessment		Verification and Validation Testing	X	X	X	X	1/19/25
Assessment GUI	Assessment Selection Page					X	1/19/25
Assessment GUI	Parent Checklist Page			X			1/22/25
Assessment GUI	Input Check Page					X	1/22/25
Assessment GUI	Assessment Questions Page		X				1/22/25
Clinician Dashboard	Result Storage Module				X		1/22/25
Assessment GUI	Assessment Instructions Page			X			1/25/25
Assessment GUI	Tutorial Page		X				1/25/25
Assessment GUI	Assessment Completion Page					X	1/25/25
Assessment GUI		Verification and Validation Testing	X	X	X	X	1/25/25
Clinician Dashboard	Report Generation Module				X		1/25/25
Clinician Dashboard		Verification and Validation Testing	X	X	X	X	1/25/25
Clinician Dashboard GUI	Clinician Dashboard Overview Page		X				1/28/25
Clinician Dashboard GUI	Patient Overview Page			X			1/28/25
Clinician Dashboard GUI	Patient Assessment Results Analysis Page					X	1/28/25
Media Processing	Media Processing Module				X		1/28/25
Clinician Dashboard GUI	Bias Review Page					X	1/31/25
Clinician Dashboard GUI	Add New Client Page		X				1/31/25
Clinician Dashboard GUI		Verification and Validation Testing	X	X	X	X	1/31/25
Homepage	Authentication Module				X		1/31/25
Homepage GUI	Select Account Type Page			X			1/31/25
Homepage GUI	Login Page (Parent) Page			X			2/3/25
Homepage GUI	Login Page (Clinician) Page			X			2/3/25
Homepage GUI	Create Account Page		X				2/3/25
Homepage GUI	Homepage (Parent) Page					X	2/3/25
Homepage GUI		Verification and Validation Testing	X	X	X	X	2/3/25
Media Processing	Video Processing Module				X		2/3/25
Media Processing	Audio Processing Module		X				2/6/25
Media Processing		Verification and Validation Testing	X	X	X	X	2/6/25
Miscellaneous	Logging Module			X			2/6/25

Miscellaneous	Real-Time Feedback Module				X		2/6/25
Miscellaneous		Verification and Validation Testing	X	X	X	X	2/6/25
Admin	Add Clinician Page					X	2/6/25
Admin		Verification and Validation Testing	X	X	X	X	2/6/25
Controllers	Landing Page GUI				X		1/19/25
Controllers		Verification and Validation Testing	X	X	X	X	1/19/25
Rev0		Full System Testing	X	X	X	X	2/8/25
Rev0		Rev0 Practice	X	X	X	X	2/9/25
Rev0		Rev0 Presentation	X	X	X	X	2/10/25

References

David L. Parnas. On the criteria to be used in decomposing systems into modules. *Comm. ACM*, 15(2):1053–1058, December 1972.

David L. Parnas. Designing software for ease of extension and contraction. In *ICSE '78: Proceedings of the 3rd international conference on Software engineering*, pages 264–277, Piscataway, NJ, USA, 1978. IEEE Press. ISBN none.

D.L. Parnas, P.C. Clement, and D. M. Weiss. The modular structure of complex systems. In *International Conference on Software Engineering*, pages 408–419, 1984.