CS445/ECE451/SE463/CS645

Software Requirements Specifications for WATTalk
# Table of Contents

1 Introduction  
   1.1 Purpose of Software Requirements Specifications  2  
   1.2 Scope of Product  2  
   1.3 Definitions, Acronyms, Abbreviations, and Notation Conventions  2  

2 Overall Description  4  
   2.1 Product Perspective  4  
   2.2 Product Features  6  
      2.2.1 Workflow Models  8  
      2.2.2 User Stories  14  
   2.3 User Characteristics  14  
   2.4 Constraints  16  
   2.5 Assumptions and Dependencies  17  

3 Specific Requirements  18  
   3.1 Domain model and UI sketches  18  
      3.1.1 Specification Domain Model  18  
      3.1.2 UI Sketches  21  
   3.2 Scenarios and State Machines  37  
      3.2.1 Scenarios  37  
      3.2.2 State Machine  46  
   3.3 Quality Requirements  51
List of Figures

1. Figure 1: Use Case Diagram ............................................. 5
2. Figure 2: UML Activity Diagram - Connect To A One-On-One Therapy Session  9
3. Figure 3: UML Activity Diagram - Access To Other Mental Health Resources . 11
4. Figure 4: UML Activity Diagram - Perform A Self-assessment ................. 12
5. Figure 5: Persona - Michael ................................................. 15
6. Figure 6: Persona - Linda ................................................... 16
7. Figure 7: Specification Domain Model Diagram .................................. 19
8. Figure 8: UI Sketch - Login Page ........................................... 21
9. Figure 9: UI Sketch - Forgot Your Password .................................. 22
10. Figure 10: UI Sketch - Student Main Page .................................. 23
11. Figure 11: UI Sketch - Therapist Main Page .................................. 24
12. Figure 12: UI Sketch - Promotion Request Form ............................... 25
13. Figure 13: UI Sketch - Quick Questionnaire ................................ 27
14. Figure 14: UI Sketch - Waiting Room ....................................... 28
15. Figure 15: UI Sketch - Meeting Room ....................................... 29
16. Figure 16: UI Sketch - External Resources Main Screen ....................... 30
17. Figure 17: UI Sketch - Our Therapists ....................................... 31
18. Figure 18: UI Sketch - Therapist Profile .................................... 32
19. Figure 19: UI Sketch - Self-assessment Main Screen .......................... 33
20. Figure 20: UI Sketch - Self-assessment Form .................................. 34
21. Figure 21: UI Sketch - Self-assessment History ................................ 35
22. Figure 22: UI Sketch - Self-assessment Record ................................ 36
23. Figure 23: Navigation Diagram ............................................... 46
24. Figure 24: Navigation Diagram Macros ........................................ 47
1 Introduction

1.1 Purpose of Software Requirements Specifications

The purpose of this document is to comprehensively define the requirements and specifications of a mental health-oriented mobile application. It is intended to provide an overview of the application’s primary functionalities from the perspective of primary use cases and the implementations required to realize said use cases.

The intended audience of this document are developers and, to a lesser extent, customers of the product. Developers may consult this document as a general reference to assist in the implementation of the application that is being described. Customers may also reference this document as a technical overview of the requirements and specifications of the application that they are using.

The readers are assumed to have basic knowledge of software requirements and specification tools such as UML, use case diagrams, domain models, and state machines.

1.2 Scope of Product

The mobile application described in this document is called WATTALK. The primary goal of the application is to provide easily-accessible mental health services to University of Waterloo students. To that end, the application allows users to obtain mental health guidance by booking one-on-one therapy sessions with registered therapists. WATTALK also provides self-assessments and contact information to other mental health-related services. The main benefit of this application is in its low barrier of entry that only a smartphone with an internet connection is required to access its services. Inconveniences of in-person therapy, such as waiting rooms and the need to travel to the therapist, are either reduced or eliminated altogether by the nature of the application.

WATTALK has limitations in that it does not eliminate the need for therapists. Rather, the application must incentivize therapists to register onto the application for its main objectives to be fulfilled. Furthermore, WATTALK is not a substitute for suicide hotlines, 911, or any other emergency service in which a person’s life or well-being is critically endangered as a result of either physical or mental conditions.

1.3 Definitions, Acronyms, Abbreviations, and Notation Conventions

- **Definitions**
  - **WATTALK**: The mobile mental health application.
  - **Application**: The application (sometimes used interchangeably with WATTALK).
  - **Unified Modeling Language**: This is a modeling language that is intended to provide a standard way to visualize the design of a system.
  - **Software Requirements Specification**: This is a document of a software system, including the descriptions of functional and non-functional requirements of the system.
  - **User Interface**: This is the space that allows users to interact with machines.
– **Users:** End users of the product. This group can be divided into two main subgroups who are uniquely catered to by the application:
  
  * **Students:** In the scope of this document, “students” are defined as university-level undergraduate or graduate students of the University of Waterloo.
  * **Therapists:** In the scope of this document, “therapists” are defined as accredited and registered therapists within the province of Ontario.

– **Login:** This refers to logging into the WATTALK application. A user logs in using a UW email and password.

– **Logout:** This refers to the action of knowingly exiting the application.

• **Acronyms**

  – **UML:** Unified Modeling Language
  – **SRS:** Software Requirements Specification
  – **UI:** User Interface
  – **App:** Application
  – **UW:** University of Waterloo
  – **MATES:** Mentor Assistance Through Education and Support
  – **CV:** Curriculum Vitae
  – **GUI:** Graphical User Interface

• **Abbreviations**

  – **App:** Application
  – **Id:** Identification
  – **Std:** Standard Deviation
2. Overall Description

This part of the documentation includes the overall description of the application, including the factors that affect the application’s requirements, the top three use cases of the application and the characteristics of users who intend to use WATTALK.

2.1 Product Perspective

The application works together with UW Portal to authenticate users and provide its services. The most important function of WATTALK is to provide online therapy sessions through a mobile platform. Other additional functions of WATTALK include performing self-assessments, accessing other mental health resources, writing in-app reviews, creating a to-do list, and joining the anonymous forum.

The following is a use case diagram that illustrates how WATTALK interacts with its actors.
The following is detailed description of each actor and use case:

- **Actor Descriptions:**
  - **UW Students**
    These are the primary human actors of WATTalk, including the University of Waterloo students at both the undergraduate and graduate levels. These students can use our application to seek mental health support by joining an online one-on-one therapy session and write reviews on our application. They also have access to other mental health resources, such as Campus Wellness and MATES.
  - **Therapists**
    The therapists are secondary human actors. When using our application, they can...
provide therapy by connecting to an online one-on-one session with the students. Further, they can also write reviews of our application.

- **University of Waterloo Portal:**
  Our application uses this non-human actor as a platform to provide online mental health services to the University of Waterloo students since our application will be embedded into the university application.

**• Use Case Descriptions**

- **Perform a self-assessment**
  Students can perform a self-assessment on their mental health status before reaching out to a therapist, and the result is saved to a history folder. Students can view answers to their past assessments. Furthermore, a graph containing the scores of all past results is accessible by students. They can also use this graph to assess mental health status over time.

- **Write reviews on the application**
  Both UW students and therapists can write reviews on our application, explicitly commenting on the mental health section.

- **Access to other mental health resources**
  In this section, websites to other external resources are accessible. These external resources include UW mental health office, off-campus therapists’ office, and UW emergency services. Moreover, the Our Therapist page contains a list of past experiences and qualifications of each UW therapist who is using the app to provide services.

- **Connect to a one-on-one therapy session**
  UW students can enter into a one-on-one secured chat message and video streaming channel with the University therapists. Each therapist can connect to one student at a time, and each student is only allowed to connect to one therapist at a time. These online therapy sessions enable students to share their current feelings and thoughts with a mental health professional who can suggest strategies to alleviate stress and anxiety.

- **Create a to-do list**
  During the therapy session, therapists may set up a to-do list for their patients before their next session. Students can also view and edit the to-do list in our application. Also, our application is capable of sending a reminder notification to students.

- **Contribute to the anonymous forum**
  Our application provides an anonymous forum for user discussion. Students can share their experiences or concerns by creating a post on the forum. In addition, therapists can also share their experiences and include some professional tips to help students manage their anxiety.

2.2 **Product Features**

This section covers the workflow diagram of the top three use cases, user stories expressing the needs and wants of each use case.

By conducting several interviews and stakeholder analysis of our most important users, the top three use cases for our application are:

- Connect to a one-on-one therapy session
2. Overall Description

- Access other mental health resources
- Perform a self-assessment

The diagrams and corresponding descriptions in this section and the following sections will solely focus on these three functions.
2.2.1 Workflow Models

The workflow model is a graphical representation of a series of tasks that are performed to accomplish some use cases. The UML activity diagram is a great way to demonstrate the workflow.

The following are the UML activity diagrams and descriptions of the top three use cases:
2. Overall Description

Description: Both university students and therapists need to login to WATTalk first. Students will then complete a quick questionnaire and request to connect to a therapist and then enter a waiting room queue. Therapists are sent directly to the meeting room after login and start accepting students in order. When the student is first in line, they

Figure 2: UML Activity Diagram - Connect To A One-On-One Therapy Session
will receive a meeting room link to join the therapy session. This link expires in five minutes, hence if they do not join within this timeframe, they will get placed at the end of the waiting room queue.

- **Elicitation techniques:**
  - Norms: a better Talkspace (i.e., an online therapy service) with faster response time and less waiting time for students at the University of Waterloo.
  - Analogical reasoning: we consider the process of joining a Zoom meeting, such as user login, enter a private meeting room.
2. Overall Description

Student Application
open WATTALK
login with waterloo credentials

[failure < 3] [failure >= 3]

click on external resources button
show main screen
show external resource page

selects our therapist button

[options = None] [options = ourTherapist]

show our therapist page

user selects therapist image

[therapistImageId = None]

show qualifications and experiences for the therapist

[therapistImageId != None]

selects our therapist button

[options = None]

Figure 3: UML Activity Diagram - Access To Other Mental Health Resources

• Description: The external resource section provides additional qualifications and experience information of each university therapist and contact information on other available resources. Contact information of other resources includes phone number, email, website, and direct personnel name if applicable. Both students and therapists could access this page.

• Elicitation techniques:
  – Brainstorm: this idea came from our group discussions
  – Constraint relaxation: other ways of providing students with mental health assistance apart from using our application.
  – Interview: students express this use case as the top use cases
Figure 4: UML Activity Diagram - Perform A Self-assessment
Description: Students can perform a new self-assessment or view previous scores and answers. The self-assessment form consists of various types of questions (e.g., multiple-choice, true or false, short answers) to determine the student’s current emotional status. On the history page, it shows a line graph displaying the assessment scores over time. The line chart displays a visual indicator of whether further assistance is needed. Furthermore, the page also provides access to past results.

Elicitation techniques:

- Stakeholder analysis: many mental health questionnaires online, and typically therapists use them as well. We thought it is a good idea to incorporate them.
- Interview: students express this use case as the top use cases
2.2.2 User Stories

User stories is a light-weight approach for managing and expressing requirements. It is a short statement that is written from the point of view of the users about any new functionality or feature.

User stories has the following structured syntax:

As a user/role, I want something, so that benefit is achieved.

The following are the user stories for the top three use cases:

- **Connect to a one-on-one therapy session**
  
  1. As a student, I want to discuss my mental health issues with a certified therapist to address my mental health concerns.
  
  2. As a therapist, I want to connect with students in need of mental health care as a part of my practice.

- **Access to other mental health resources**
  
  1. As a student, I want a single place where I can access various mental health resources such as therapist phone numbers and emergency contacts.
  
  2. As a student, I want to be able to access alternate mental health resources in the event that I am not able to directly contact a therapist.

- **Perform a self-assessment**
  
  1. As a student, I want to perform a psychological self-assessment so that I may better gauge my mental health needs.

  2. As a student, I want to quantify my general mental well-being through metrics that are easily understandable.

2.3 User Characteristics

Users of WATTALK are required to have a valid UW email and have sufficient knowledge about using a mobile phone (e.g., installing applications on mobile phones, performing online chat and sending messages through mobile phones).

The following are the two main users of WATTALK:

- **UW students**
  
  These are users that are seeking mental health support and are studying at the University of Waterloo, including undergraduate and graduate students.

- **Therapists**
  
  These are licensed mental health professionals that specializes in college student mental health problems. They are assumed to be able to provide online therapy through mobile phones.

On the other hand, personas are useful when real users are not reachable or are too numerous to interview them all. Personas are invented personalities but they can be used to elicit the
requirements and specifications of the software or product.

The following are two personas for WATTALK:

**Problems and pain points**
Michael is new to the University of Waterloo’s working environment, but is trying to acclimate himself to the best of his ability. As a new hire, he strongly feels that he needs to produce substantial achievements - that is, successful production projects - in order to prove himself. In spite of this, however, his most recent WATTALK integration initiative suffers from various issues.

These issues stem from the lack of consistent communication between the WATTALK and Portal development teams, as they are not in direct contact with each other. It is therefore sometimes unclear as to what is required of the UWaterloo Portal side of development, and whether certain features should exist in either WATTALK or Portal.

**Goals**
- Successfully integrate UWaterloo Portal with WATTALK by providing links and prompts to WATTALK via the UWaterloo Portal interface
- Earn the commendation of his superiors at work
- Long-term: become a senior or lead developer within two to three years

**Bio**
Michael is a 28 year-old junior software engineer working for the University of Waterloo. At work, his main project is the University of Waterloo Portal (Portal for short).

Michael has roughly five years of experience in developing web applications in a professional environment. He recently joined the Portal development team. Before coming to the university, he worked at Maplesoft for some period of time. His greatest areas of software expertise are in user interface design and data protection.

Michael’s most recent work on Portal concerns the integration of WATTALK. As a university-endorsed application, WATTALK must be integrated onto the UWaterloo Portal such that the former is accessible via the through links and prompts within the latter.

**Figure 5: Persona - Michael**
2.4 Constraints

- **Regulatory policies**
  - All therapists must be certified and qualified to perform therapy.

- **Hardware limitations**
  - Non-verbal body language may be missing because the therapists can only see the face of students.
  - Notification must be enabled for students to receive meeting links.
  - Some therapists may request a video call but the student may not be on a device with a working camera.

- **Parallel operations**
  - WATTALK should be able to run as background application in the mobile, students should still be in queue even if they are browsing another application.

- **Audit functions**
  - Checks for data integrity and identity protection.

- **Control functions**
  - Both the therapist and students must be able to control the audio/video of themselves in a session.

- **Criticality of application**
2. Overall Description

Data leakage is the only major concern of the application.

- **Safety and security considerations**
  - The contents of the call (video and audio) should not be leaked or stored anywhere in the application.
  - The self-assessment scores and results must be behind several layers of encryption.

- **Laws**
  - PIPEDA must be followed to protect student confidential information

2.5 Assumptions and Dependencies

When building a system, it is nearly impossible that by itself will satisfy all the requirements. Therefore, assumptions about how the environment behaves can be used.

The following are the assumptions of the top three use cases for WATTALK:

- **General assumptions**
  - Students and therapists remember their ID and password.
  - Students and therapists have a mobile phone with a touch screen or a keyboard.
  - Students and therapists have a mobile phone that contains a camera, microphone, and has internet access.
  - Students and therapists have installed the application on their mobile phone.
  - If the mobile phone camera is enabled, it is able to capture users’ faces.
  - If the mobile phone microphone is enabled, it is able to capture users’ voice.

- **Assumptions for connect to a one-on-one therapy session**
  - General assumptions specified above.
  - Students provide honest answers to the questionnaire.
  - The matching algorithm correctly finds the therapist based on the student’s needs.
  - The therapist found by the algorithm is active and available.
  - There exists a therapist that covers student’s needs.
  - Students and therapists can communicate with each other through either messages or voices in the meeting room.
  - The therapist and the student are using the meeting room simultaneously.

- **Assumptions for perform a self-assessment**
  - General assumptions specified above.
  - Students provide honest answers to the self-assessment questions.
  - A record of students’ self-assessment responses is kept in the application database.

- **Assumptions for access to other mental health resources**
  - General assumptions specified above.
  - Therapists fill in the promotion request form honestly.
3 Specific Requirements

This section describes the requirements and specifications for WATTALK in detail. It contains the interface specification including domain models and UI sketches, scenarios, state machine models, and quality requirements.

3.1 Domain model and UI sketches

This section contains the detailed specification domain model and UI sketches of the three top-priority use cases for WATTALK, which are connecting to a one-on-one therapy session, performing self-assessment, and accessing external resources.

The specification domain model is shown in Figure 7, including the human actors (i.e., UW students, therapists), non-human actors (i.e., UW Portal), interfaces and other elements of the system, and associations between them.

The UI sketches are shown from Figure 8 to Figure 22, containing the UI elements (e.g., buttons, text boxes, icons) for each screen, draft layout, and also the event that could be used as transitions in the navigation diagram, as shown in Figure 23.

3.1.1 Specification Domain Model

Below is specification domain model diagram for WATTALK:
Below are the descriptions of previous specification domain model diagram, including entity descriptions and relation descriptions.

- **Student Account**: account belongs to student, contains basic information required for login and session related summary (agenda).

- **Therapist Account**: account belongs to therapist, contains basic information required for login and their schedules.

- **Therapist Promotional Profile**: contains a complete list of the therapist’s information (e.g., Curriculum Vitae, award, certificates), and therapists can create this profile by filling a promotion request form, and update their profiles by filling multiple promotion request forms.
– **University of Waterloo Portal**: a service of University of Waterloo to authenticate student and therapist accounts.

– **External Resource**: contains a list of all the available therapists’ resources. Promotion Request Form: a form that a therapist fills in to request to create a complete profile and add it to the external resources to promote their services.

– **Self-assessment Form**: a form that students fill in to evaluate their current mental health condition.

– **Self-assessment Folder**: contains a history of all the self-assessment forms and results that a student has filled in.

– **Self-assessment Record**: a record for each submitted form containing the form number, the answers for each question, submitted date and final result.

– **Quick Questionnaire**: a short questionnaire that students fill in to specify their special concerns for the following therapy session.

– **Notification Center**: a place for students and therapists to check their notifications which direct them to the meeting link.

– **Queue/Waiting Room**: a room for students waiting for the therapist where they are notified of their number in line and when they are up next.

– **Meeting Room**: a room for one student and one therapist to engage in receiving and providing therapy.

*Relation Descriptions*

– **General relation**
  * A student can create only one student account.
  * A therapist can create only one therapist account.
  * A therapist can create only one complete profile.
  * A therapist can edit their respective profile.
  * A student can view the therapist profiles’ name, occupation, email, schedule and office locations.
  * A therapist can view the student profiles’ name, occupation, email, and agenda.
  * University of Waterloo Portal authenticates both the student and the therapist accounts.
  * The notification center can send multiple notifications to both students and therapists.

– **Access to other mental health resources**
  * An external resource list can contain multiple therapist profiles.
  * A therapist can complete multiple promotion request forms to update the profile.
  * A student can view multiple resources.

– **Perform a self-assessment**
  * A self-assessment folder can contain multiple self-assessment forms, their respective results and the time submitted.
  * A self-assessment is completed by one student.
  * A student can complete multiple self-assessment forms, and all the forms completed will be saved in the self-assessment folder.
  * A student can have only one self-assessment folder.
3. Specific Requirements

- A self-assessment folder can contain multiple self-assessment records that contain the form number, answers for each question, submitted date and final result for each submitted form.

  - **Connect to a one-on-one therapy session**
    - A meeting room can have at most one student.
    - A meeting room can have at most one therapist.
    - A student fills out a single quick questionnaire before each therapy session.
    - A queue/waiting room can have multiple students.
    - A therapist can be assigned to one queue/waiting room at a time.
    - A student can be in one waiting room at a time.
    - A student/therapist can be in one meeting room at a time.

3.1.2 UI Sketches

The following are UI sketches for WATTALK:

![Login Page GUI Components](image)

**Figure 8: UI Sketch - Login Page**

**Login Page GUI Components:**

- **WATTALK**
  - Application logo and name

- **Username input field**
  - The username must be the email created by the University of Waterloo

- **Password input field**
3. Specific Requirements

- This is the password that is associated with user’s email used on all Waterloo Platforms (ex: Learn, Quest)

• **Forgot your password? button**
  - Clicking on this button will open a prompt you to input your waterloo email to retrieve/reset your University of Waterloo email password

• **Sign in button**
  - Clicking on this button will sign you into the application using the credentials provided in the User input field and the Password input field (go to Main Scree (as shown in Figure 10/11) on success for student/therapist)
  
  \[Events Triggered: Login(username, password)\]

• **University of Waterloo logo**
  - The logo represents that WATTALK is affiliated with the University of Waterloo

![UI Sketch - Forgot Your Password](image)

**Figure 9: UI Sketch - Forgot Your Password**

**Forgot Your Password Page GUI Components:**

• **User email input field**
  - Users need to enter their UW email for resetting the password.

• **New password input field**
  - This is the new password that will be associated with the user’s email that is used to login to the application.

• **Confirm password input field**
Repeat to enter the new password for confirmation.

- **Back button**
  - Clicking the back button will return the user to the login page (as shown in Figure 8)

- **Submit button**
  - Clicking the submit button will update the associated password based on the user input email and new password, and return the user to the login page. (as shown in Figure 8)

![Figure 10: UI Sketch - Student Main Page](image)

Student Main Page GUI Components:

- **Log out button**
  - Clicking on the button will log the student out from their account and return to the Login Page (as shown in Figure 8)
  
  *Events Triggered: logout()*

- **Welcome! sign**
  - A static sign indicating that the student is on the main page of the application

- **Account button**
  - Clicking on the account button will redirect the student to their account page

- **Setting button**
Clicking on the button will redirect the student to their settings page

- **Connect to one-on-one therapy session button**
  - Clicking on this button will display a pop-up, prompting the student to fill in a quick questionnaire
    
    _Events Triggered: clickTherapySession()_

- **Perform a self-assessment button**
  - Clicking on this button will redirect the student to the self-assessment page
    
    _Events Triggered: clickSelfAssessmentForm()_

- **External resources button**
  - Clicking on this button will redirect the student to the external resources page
    
    _Events Triggered: clickExternalResources()_

---

**Figure 11: UI Sketch - Therapist Main Page**

**Therapist Main Page GUI Components:**

- **Log out button**
  - Clicking on the button will log the therapist out from their account and return to the Login Page (as shown in Figure 8)
    
    _Events Triggered: logout()_

- **Welcome! sign**
  - A static sign indicating that the therapist is on the main page of the application

- **Account button**
3. Specific Requirements

- Clicking on the account button will redirect the therapist to their account page

- **Setting button**
  - Clicking on the button will redirect the therapist to their settings page

- **Connect to one-on-one therapy session button**
  - Clicking on this button will redirect the therapist to the meeting room page
  
  *Events Triggered: `clickTherapySession()`*

- **Promotion request form button**
  - Clicking on this button will redirect the therapist to a form which they can fill out to include their qualifications and experiences

  *Events Triggered: `clickPromotionForm()`*

- **External resources button**
  - Clicking on this button will redirect the therapist to the external resources page

  *Events Triggered: `clickExternalResources()`*

![Figure 12: UI Sketch - Promotion Request Form](image)

Promotion Request Form Page GUI Components:

- **Log out button**
  - Clicking on the button will log the therapist out from their account and return to the Login Page (as shown in Figure 8)

- **Name input Field**
3. Specific Requirements

- Therapist types their full name

- **Curriculum Vitae field**
  - The CV field will display the name of the file uploaded using the upload file button adjacent to the CV field

- **Certificate field**
  - The certificate field will display the name of the file uploaded using the upload file button adjacent to the Certificate field

- **Award field**
  - This field will display the name of the file uploaded using the upload file button adjacent to the Award field

- **upload file button**
  - Will prompt the therapist to choose a file to upload to the application

- **Back button**
  - Clicking the back button will return the therapist to the main page
    "Events Triggered: back()"

- **Submit button**
  - Clicking on the submit button will send this data to WATTALK server, returning the therapist to the main screen
    "Events Triggered: promotionSubmit()"

- **“Scroll Wheel”**
  - The scroll wheel indicates how far down on the page you are
  - Using the scroll wheel will take you to different parts of the page

- **Lines under Award field**
  - Indicates that the promotion request form contains more fields such as Achievements field that are similar to those fields shown in the UI screen. These unshown fields are omitted because of space reasons.
Quick Questionnaire Page GUI Components:

- Log out button
  - Clicking on the button will log the student out from their account and return to the Login Page (as shown in Figure 8)
    
    *Events Triggered: logout()*

- Nickname input field
  - Student can put a preferred name in this field, else the name inputted when making the account will be used.

- Do you have any particular concern input field
  - Student can input reason for requesting therapist.

- Do you have any preferred therapist? dropdown
  - Clicking the button gives a dropdown including the names of all available therapists and the option of “any”

- Skip button
  - Clicking this button puts the student in a waiting room with no information attached to their profile.
    
    *Events Triggered: questionnaireSkip()*

- Submit button
  - Clicking this button will attach this information temporarily to the student profile and put them into the waiting room
    
    *Events Triggered: questionnaireSubmit()*
Waiting Room Page GUI Components:

- **Log out button**
  - Clicking on the button will log the student out from their account, losing their position in the queue, and return to the Login Page
    
    \textit{Events Triggered: logout()}

- **Number indicating queue position**
  - This number will decrease (as low as 1) as students get paired with a therapist

- **Cancel button**
  - Clicking this button will remove the student from the queue and redirect them to the main page
    
    \textit{Events Triggered: leaveWaitingRoom()}
Figure 15: UI Sketch - Meeting Room

### Meeting Room Page GUI Components:

- **Microphone icon**
  - Clicking this icon will mute/unmute the audio of the users’ device

- **Video icon**
  - Clicking the video icon will enable/disable video on the users’ device

- **End button**
  - Clicking this button will end the call for both parties
  
  *Events Triggered: endCall()*

- **Chat icon**
  - Clicking this icon will open/close a chat room that only members within the meeting room can access.
External Resources Main Screen GUI Components:

- Log out button
  - Clicking on the button will log the user out from their account and return to the Login Page
    
    *Events Triggered: logout()*

- Our therapists button
  - Clicking on this button will redirect to another page dedicated to therapists working under the University of Waterloo
    
    *Events Triggered: clickOurTherapists()*

- External website Links dropdown
  - Clicking on this button will open a dropdown which includes links to external mental health resources/websites
  - Clicking on any of the links will temporarily close the app and redirect the user to the website using the default web browser on the device

- Report broken Link dropdown
  - Clicking on this button will open a dropdown which includes all the links provided under the External website Links dropdown
  - Select all links that are broken by clicking them
  - Clicking the Submit button will send all the links to be checked, removed and replaced

- Back button
– Clicking on the back button will return the user to their main screen.

*Events Triggered: back()*

![Our therapists GUI Components](image)

**Figure 17: UI Sketch - Our Therapists**

**Our Therapists GUI Components:**

- **Log out button**
  
  – Clicking on the button will log the user out from their account and return to the Login Page
  
  *Events Triggered: logout()*

- **Face icon**
  
  – Clicking on any of the therapist face icons will open a read-only pop-up with some information on the therapist chosen
  
  – Clicking on “view more” button will redirect the user to another page which displays the full profile of the chosen therapist

- **Back button**
  
  – Clicking the back button will return the user to the external resources page
  
  *Events Triggered: back()*

- **View more button**
  
  – Clicking the View more button will direct the user to the specific therapist’s promotional profile page
  
  *Events Triggered: viewMore(Therapist.Email)*

- **“Scroll Wheel”**
The scroll wheel indicates how far down on the page you are

Using the scroll wheel will take you to different parts of the page

**Figure 18: UI Sketch - Therapist Profile**

**Therapist Profile GUI Components:**

- **Log out button**
  - Clicking on the button will log the user out from their account and return to the Login Page
  
  *Events Triggered: logout()*

- **Face icon**
  - This represents the therapist’s uploaded photo when they fill the promotional request form.

- **Background Photo display field**
  - This field represents the background photo uploaded by the therapist when they fill the promotional request form.

- **Static fields** (e.g., Name, Office, Phone, Website, Certificates, Award)
  - This field represents the corresponding image or input text submitted by the therapist when they fill the promotional request form.

- **Lines under Award field**
  - Indicates that the therapist promotional profile contains more fields such as Achievements field that are similar to those fields shown in the UI screen. These unshown fields are omitted because of space reasons.
• Back button

  – Clicking on this button will redirect the user back to the Our therapist page

  \textit{Events Triggered: back()}

\begin{figure}[h]
\centering
\includegraphics[width=0.5\textwidth]{fig19.png}
\caption{UI Sketch - Self-assessment Main Screen}
\end{figure}

\textbf{Self-assessment Main Screen GUI Components:}

• Log out button

  – Clicking on the button will log the student out from their account and return to the Login Page

  \textit{Events Triggered: logout()}

• Submit a self-assessment form button

  – Clicking on this button will redirect the student to another page which is the self-assessment form

  \textit{Events Triggered: clickSelfAssessmentForm()}

• View self-assessment history button

  – Clicking on this button will redirect the student to another page which displays every self-assessment they have filled out

  \textit{Events Triggered: clickSelfAssessmentHistory()}

• Back button

  – Clicking on this button will redirect the student back to the main page

  \textit{Events Triggered: back()}
Self-assessment Form GUI Components:

- **Log out button**
  - Clicking on the button will log the student out from their account and return to the Login Page
  
  *Events Triggered: logout()*

- **Option X button** where X is a number starting at 1
  - Clicking on this button will save your response for this from

- **“Rate the following statements” Question**
  - Clicking on a number from 1 to 5 will save the value clicked on this form

- **Back button**
  - Clicking the button will return the student back to the self-assessment page
  
  *Events Triggered: back()*

- **Submit button**
  - Clicking this button will save the form, displaying a success message and return the user back to the self-assessment page
  
  *Events Triggered: selfAssessmentSubmit()*

- **“Scroll Wheel”**
  - The scroll wheel indicates how far down on the page you are
  - Using the scroll wheel will take you to different parts of the page
Self-assessment History GUI Components:

- **Log out button**
  
  - Clicking on the button will log the student out from their account and return to the Login Page
  
  *Events Triggered: logout()*

- **Form X** where X is a number starting from 1
  
  - Clicking on any of the forms within the form column will display the questionnaire and the corresponding responses on the same page
  
  *Events Triggered: clickFormNum(formId)*

- **Back button**
  
  - Clicking on the back button will return the student to the self-assessment page
  
  *Events Triggered: back()*
Self-assessment Record GUI Components:

- **Log out button**
  - Clicking on the button will log the student out from their account and return to the Login Page
  
  *Events Triggered: logout()*

- **Header display field**
  - Contains the form number, submit date, and the final score.

- **Question and answer display field**
  - Contains the form question and user answers.

- **Back button**
  - Clicking on the back button will return the student to the self-assessment history page
  
  *Events Triggered: back()*

- **“Scroll Wheel”**
  - The scroll wheel indicates how far down on the page you are
  - Using the scroll wheel will take you to different parts of the page
3.2 Scenarios and State Machines

This section contains the scenarios and state machine of the top three use cases.

Scenarios represent one full observable actions execution path through a use case, including the interactions between the system and any external actors.

For WATTALK, that’s the interactions between the system and the user (e.g., students, therapists), and also include the actions between the users.

On the other hand, a state machine model represents the system’s behavior, and it could serve as an navigation diagram where transitions are triggered by input events and the UI screens are displayed as states in the model.

3.2.1 Scenarios

The following section includes scenarios and sub-scenarios pertaining to our top three use cases for WATTALK, those being: Connect to a one-on-one Therapy Session, Perform a Self-assessment and Access to external resources.

Sub-scenario: Login
### 3. Specific Requirements

<table>
<thead>
<tr>
<th>User</th>
<th>WATTALK</th>
<th>Database</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Enters username and password, clicks on the Sign in button</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2. Sends username and password to database for authentication</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3. Authenticate password based on entered username</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4. Returns true and successful login token</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5. Changes view from login to main screen</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Alternative 1: Failed login**

<table>
<thead>
<tr>
<th></th>
<th>A1.4. Returns false indicating unsuccessful login</th>
</tr>
</thead>
<tbody>
<tr>
<td>A1.5. Displays incorrect username or password message to user and increment attempt counter</td>
<td></td>
</tr>
<tr>
<td>A1.6. Go back to step 1</td>
<td></td>
</tr>
</tbody>
</table>

**Alternative 2: Forgot password**

<table>
<thead>
<tr>
<th></th>
<th>A2.2. Enters email to receive link to reset password</th>
</tr>
</thead>
<tbody>
<tr>
<td>A2.3. Sends link to reset password to provided email</td>
<td></td>
</tr>
<tr>
<td>A2.4 Student receives email, clicks on link to reset password</td>
<td></td>
</tr>
<tr>
<td>A2.5 update password</td>
<td></td>
</tr>
<tr>
<td>A2.6 go back to step 1</td>
<td></td>
</tr>
</tbody>
</table>

**Exception 2: Internet fails**

<table>
<thead>
<tr>
<th></th>
<th>E2.2. Displays connection error and ask users to try again later</th>
</tr>
</thead>
</table>

**Exception 3: Server error, cannot connect to database**

| | E3.2. Go to Exception EX |

**Sub-scenario: Exception EX**
### Scenario 1

**Use Case Name:** Connect to a one-on-one Therapy Session  
**Trigger:** Student clicks on “connect to one-on-one therapy session” button  
**Preconditions:** quest app downloaded, user is UW student and logged in, currently viewing the main page  
**Stakeholders:** therapists, undergraduate and graduate UW students  
**Post Conditions:** student leaves meeting room after receiving therapy

<table>
<thead>
<tr>
<th>Student</th>
<th>WATTALK</th>
<th>Database</th>
<th>Therapist</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Login with credentials using login sub scenario</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2. Clicks on Connect to one-on-one therapy session button in main screen</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3. Displays short questionnaire for student to complete</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4. Clicks Submit</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5. Save student answers in database under their profile and puts student in waiting room</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6. Student enters waiting room and is notified when they are first in line</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7. Retrieves agenda, quick questionnaire and self-assessment results of student from database of student first in line</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Step</td>
<td>Description</td>
<td></td>
<td></td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3. Specific Requirements CS445/ECE451/SE463/CS645</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8. Receives “agenda” notes on the student, quick questionnaire and self-assessment results</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9. Creates a meeting room that therapist is waiting in and notifies student to join it</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10. Clicks on link provided by application and joins video conference room</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>*11. Therapy Session begin, student and therapist begin to communicate with each other</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12. Ends the video conference by pressing the End button, closing the video conference</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Alternative 1: Student exit waiting room</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A1.6 The student leaves the waiting room by pressing Cancel</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A1.7 Remove student from the waiting room</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A1.8 Redirects student to home page</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Alternative 2: Student doesn’t click on the link</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A2.9 Link expires after 5 minutes</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A2.10 Go to 5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Alternative 3: Therapist ends their day early</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A3.1 Therapist clicks Log out</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A3.2 Clears waiting room and notifies all students</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A3.3 Go to 5</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Alternative 4: No Therapists online

<table>
<thead>
<tr>
<th>Event</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>A4.3</td>
<td>Message Appears stating no Therapists are online</td>
</tr>
<tr>
<td>A4.4</td>
<td>Student closes message</td>
</tr>
</tbody>
</table>

### Alternative 5: Skips questionnaire

<table>
<thead>
<tr>
<th>Event</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>A5.5</td>
<td>Clicks “skip” button</td>
</tr>
<tr>
<td></td>
<td>A5.6 Go to step 6</td>
</tr>
</tbody>
</table>

### Exception 1: Quick questionnaire responses are not saved

<table>
<thead>
<tr>
<th>Event</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>E1.5</td>
<td>Quick questionnaire responses are not stored in the database and user is notified of the error</td>
</tr>
<tr>
<td>E1.6</td>
<td>Notify student responses are not saved due to database error</td>
</tr>
<tr>
<td>E1.7</td>
<td>Go to step 3</td>
</tr>
</tbody>
</table>

### Exception 2: Student gets booted out of the waiting room due to timeout

<table>
<thead>
<tr>
<th>Event</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>E2.1</td>
<td>User gets booted out of the waiting room due to timeout and notified of the error</td>
</tr>
<tr>
<td>E2.2</td>
<td>Go to 5</td>
</tr>
</tbody>
</table>

### Exception 3: Student’s notes are not sent to the therapist

<table>
<thead>
<tr>
<th>Event</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>E3.1</td>
<td>The error is displayed to the student and the therapist</td>
</tr>
<tr>
<td>E3.2</td>
<td>Go to 6</td>
</tr>
</tbody>
</table>

*The order of the events of the same number does not matter.*

### Scenario 2

**Use Case Name:** Perform a Self-assessment  
**Trigger:** Student click on “perform a self-assessment” button  
**Preconditions:** quest app downloaded, user is waterloo student and logged in, currently viewing the main page  
**Stakeholders:** undergraduate and graduate UW students  
**Post Conditions:** student sees completed form under “self-assessment history” page with completed date and time
<table>
<thead>
<tr>
<th>Student</th>
<th>WATTALK</th>
<th>Database</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Login with credentials using login sub scenario</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4. Student clicks on Submit a self-assessment form button</td>
<td>5. Displays self-assessment page and makes request to database to get questions</td>
<td>6. Retrieves self-assessment questions and send these records back to WATTALK</td>
</tr>
<tr>
<td>8. Student responds to questions and clicks Submit</td>
<td>9. WATTALK sends the student responses to database and account ID</td>
<td>10. Adds the new student’s answers and timestamp to the self-assessment portion</td>
</tr>
<tr>
<td>Alternative 1: Mandatory questions are incomplete</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A1.8. Displays need to finish all mandatory questions before proceeding message to student</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A1.9. Go to Step 7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Alternative 2: Student selects ‘back’ button</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A2.8. Student clicks Back button</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A2.9. Go back to step 3 and wait for student input</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Alternative 3: Reloads questions</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A4.6. Display questions and reload temporarily saved answers from phone</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A4.7. Go to Step A1.7</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Alternative 4: View self-assessment history

<table>
<thead>
<tr>
<th>A4.4. Click on View self-assessment history button</th>
<th>A4.5. Display previous answer page and requests previous answers from database</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>A4.6. Return all previous self-assessment answers based on date and time</td>
</tr>
</tbody>
</table>

A4.7. Generates a scatter plot summarizing all previous self-assessment results

A4.8. Display scatter plot at the top of the “Self-assessment history” page

A4.9. Display each set of answers in a table, which includes the file name and time of completion

A4.10. Student selects an answer set

Exception 1: Internet Fails to Communicate with Server

E1.A4.5. Times out waiting for internet to send request to view the self-assessment history

Exception 2: Database Connection Error

E2.5. Refer to sub-scenario Exception 1

Scenario 3

Use Case Name: access to external resources

Trigger: student click on External resources button

Preconditions: quest app downloaded, user is waterloo student and logged in, currently viewing the main page

Stakeholders: undergraduate and graduate UW students, UW therapists

Post Conditions: Student gains access to provided external resources

<table>
<thead>
<tr>
<th>Student</th>
<th>WATTALK</th>
<th>Database</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Login with credentials using login sub-scenario</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2. Clicks on External resources button</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>3.</td>
<td>Displays external resource main page and request for available resources stored in database</td>
<td>4. Returns a list of available resources: “Our Therapists”, “External Website Links”</td>
</tr>
<tr>
<td>5.</td>
<td>Displays therapist names and external website links in 2 separate list formats</td>
<td></td>
</tr>
<tr>
<td>6.</td>
<td>Clicks on External website links button</td>
<td>7. Display a dropdown list of all external websites</td>
</tr>
<tr>
<td>8.</td>
<td>Click on a single link</td>
<td>9. Opens default browser with link selected</td>
</tr>
<tr>
<td>Alternative 1: Choose a Therapist Profile to View</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A1.6</td>
<td>Click on Our therapists button</td>
<td>A1.7 Redirects to “Our therapists” page</td>
</tr>
<tr>
<td>Alternative 1.1 Exit “Our Therapist Page”</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A1.1.8</td>
<td>Student can hit Back button on their phone to return to External Resources page</td>
<td></td>
</tr>
<tr>
<td>A1.2</td>
<td>Choose a Therapist</td>
<td></td>
</tr>
<tr>
<td>A1.2.8</td>
<td>Student clicks on the face of a therapist</td>
<td>A1.2.9 Small preview of therapist profile information shown</td>
</tr>
<tr>
<td>A1.2.1</td>
<td>Click View More</td>
<td></td>
</tr>
<tr>
<td>A1.2.1.10</td>
<td>Student clicks view more button</td>
<td>A1.2.1.11 Displays Therapist Profile page populated with information about selected therapist</td>
</tr>
<tr>
<td>A1.2.2</td>
<td>Choose Another Therapist</td>
<td></td>
</tr>
<tr>
<td>A1.2.2.10</td>
<td>Student continuously selects different therapist profiles, while staying on the “Our Therapists” page</td>
<td></td>
</tr>
<tr>
<td>Exception 1: Database Error</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Exception 2: Opening external app error

<table>
<thead>
<tr>
<th>E1.3. Refer to Exception 1</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>E2.7. Displays error stating unable to open app, ask user to try again later</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>E2.8. Stays on same page</th>
</tr>
</thead>
</table>

### Exception 3: Report website link

<table>
<thead>
<tr>
<th>E3.8. Clicks on Report broken Link button</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>E3.9. Displays a dropdown of all websites are shown</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>E3.10. Choose the website(s) link that are broken and click Submit</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>E3.11. Sends report to database indicating that website(s) link are broken</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>E3.12. Creates log of the report</th>
</tr>
</thead>
</table>
3.2.2 State Machine

Below is the navigation diagram and the legend descriptions for WATTALK of the top three use case, the states in the navigation diagram is the UI screens, and the transitions are triggered by input events (e.g., button clicks).

![Navigation Diagram](image)

**Figure 23: Navigation Diagram**

Below is the macros for the navigation diagram:
Variable macros
userOcc = user.Occupation
W.capacity = WaitingRoom.Capacity
isFirst = WaitingRoom.isFirst
M.capacity = MeetingRoom.Capacity
M.Therapist = MeetingRoom.TherapistNum
M.student = MeetingRoom.StudentNum
Email = user.Email
OOS = OneOnOneSession
ER = ExternalResource
S.QueuePos = StudentAccount.QueuePos

Event macros
clickTS() = clickTherapySession()
clickER() = clickExternalResources()
clickPF() = clickPromotionForm()
clickPSA() = clickPerformSelfAssessment()
quesSm() = questionnaireSubmit()
quesSk() = questionnaireSkip()
leaveWR() = leaveWaitingRoom()
clickOT() = clickOurTherapists()
clickSAF() = clickSelfAssessmentForm()
SAFSubmit() = selfAssessmentSubmit()
clickSAH() = clickSelfAssessmentHistory()
promoSm = promotionSubmit()

---

Figure 24: Navigation Diagram Macros

Legend 1:

<table>
<thead>
<tr>
<th>States</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Login page</td>
<td>A page that both students and therapists see when they first open the application, as shown in Figure 8.</td>
</tr>
<tr>
<td>Student main page</td>
<td>A page that students see after a successful login contains buttons that let students connect with a therapist, fill a self-assessment form, check out external resources or logout, as shown in Figure 10.</td>
</tr>
<tr>
<td>Therapist main page</td>
<td>A page that therapists see after a successful login contains buttons that let therapists connect to a patient, fill out a promotion request form, check out external resources or logout, as shown in Figure 11.</td>
</tr>
<tr>
<td>Quick questionnaire</td>
<td>A questionnaire that students fill out just before connecting to a therapy session to give the therapist an overview of the student’s current mental health state, as shown in Figure 13.</td>
</tr>
<tr>
<td>Waiting room</td>
<td>A room where students can wait for their turn to connect to a therapist. They are notified of their place in line, as shown in Figure 14.</td>
</tr>
<tr>
<td>Meeting room</td>
<td>A room where students and therapists can connect virtually and have their virtual therapy session, as shown in Figure 15.</td>
</tr>
<tr>
<td>External resource main page</td>
<td>A page where students and therapists can access external mental health help resources, as shown in Figure 16.</td>
</tr>
<tr>
<td>Our Therapists page</td>
<td>A page where students and therapists can view the bios of all the therapists that offer virtual therapy sessions on WATTALK, as shown in Figure 17.</td>
</tr>
<tr>
<td>Self-assessment main page</td>
<td>A page that contains buttons to let students finish a self-assessment form or view their self-assessment form history, as shown in Figure 16.</td>
</tr>
<tr>
<td><strong>Self-assessment form</strong></td>
<td>A page that contains a series of questions (e.g., visual, multiple choice) for students to fill out so the therapist can get a deeper understanding of the student’s mental health history and state, as shown in Figure 20.</td>
</tr>
<tr>
<td>-------------------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>Self-assessment history</strong></td>
<td>A page that contains all the past self-assessment form’s final scores so the student can recognize their change in mental health state throughout using the application, as shown in Figure 21.</td>
</tr>
<tr>
<td><strong>Self-assessment record</strong></td>
<td>A page that contains a specific instance of a previously filled out self-assessment form so the student can view their exact answers to specific questions, as shown in Figure 22.</td>
</tr>
<tr>
<td><strong>Promotion request form</strong></td>
<td>A page that displays promotional form enabling therapists to enter their experiences and qualifications, as shown in Figure 12.</td>
</tr>
<tr>
<td><strong>Therapist Promotional Profile</strong></td>
<td>Displays the experience and qualifications of the requested therapist, as shown in Figure 18.</td>
</tr>
</tbody>
</table>

**Legend 2:**

<table>
<thead>
<tr>
<th><strong>Transitions</strong></th>
<th><strong>Description</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>login()</td>
<td>User initiates login (user must enter login information first). If user information is authenticated, then go to the user main page (e.g., student main page or therapist main page).</td>
</tr>
<tr>
<td>logout()</td>
<td>User initiates logout via the “logout” button</td>
</tr>
<tr>
<td>back()</td>
<td>User returns to previous screen/state</td>
</tr>
<tr>
<td>clickTherapySession()</td>
<td>User clicks the button to initiate one-on-one Therapy Session (from Main Page)</td>
</tr>
<tr>
<td>clickExternalResources()</td>
<td>User clicks the button to go to External Resources main page (from Main Page)</td>
</tr>
<tr>
<td>questionnaireSkip()</td>
<td>User clicks the button to skip pre-Therapy Session questionnaire and go to Waiting Room</td>
</tr>
<tr>
<td>questionnaireSubmit()</td>
<td>User clicks the button to submit the pre-Therapy Session questionnaire and go to Waiting Room</td>
</tr>
<tr>
<td>leaveWaitingRoom()</td>
<td>User clicks the button in the waiting room to cancel the current queued meeting and return to Main Page</td>
</tr>
<tr>
<td>endCall()</td>
<td>User clicks the button to terminate the current one-on-one Therapy Session call</td>
</tr>
<tr>
<td>clickOurTherapists()</td>
<td>User clicks the button to enter “Our Therapists” page (from External Resources main page)</td>
</tr>
<tr>
<td>clickSelfAssessmentForm()</td>
<td>User clicks button to initiate the process of a Self-Assessment Form (from Self-Assessment Main Page)</td>
</tr>
<tr>
<td>selfAssessmentSubmit()</td>
<td>User clicks the button to submit Self-Assessment Form</td>
</tr>
<tr>
<td>clickSelfAssessmentHistory()</td>
<td>User clicks the button to view Self-Assessment History (from Self-Assessment Main Page)</td>
</tr>
<tr>
<td>clickFormNum(formId)</td>
<td>User clicks the button to view specific Self-Assessment result (from Self-Assessment History)</td>
</tr>
<tr>
<td>clickPromotionForm()</td>
<td>User (Therapist) clicks on this button to open the Promotion Form.</td>
</tr>
<tr>
<td>clickPerformSelfAssessment()</td>
<td>User (Therapist) clicks on this button to open the Promotion Form.</td>
</tr>
</tbody>
</table>
3. Specific Requirements

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>clickPerformSelfAssessment()</td>
<td>User clicks the button to go to the Self-Assessment home page.</td>
</tr>
<tr>
<td>promotionSubmit()</td>
<td>User (Therapist) submits a promotion form request to have view more information updated on Our Therapist page.</td>
</tr>
<tr>
<td>viewMore(Therapist.Email)</td>
<td>Shows experiences and qualifications of therapist selected.</td>
</tr>
<tr>
<td>updatePos()</td>
<td>This function updates the position of all students who are in the waiting room.</td>
</tr>
</tbody>
</table>

**Legend 3:**

<table>
<thead>
<tr>
<th>Conditions/Events</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>StudentAccount.Email = username &amp;&amp; StudentAccount.Password = hash(password)</td>
<td>When students try to log in to the application, their entered username and password will be compared with the record in the database for authentication.</td>
</tr>
<tr>
<td>TherapistAccount.Email = username &amp;&amp; TherapistAccount.Password = hash(password)</td>
<td>When therapists try to log in to the application, their entered username and password will be compared with the record in the database for authentication.</td>
</tr>
<tr>
<td>userOcc = therapist</td>
<td>User occupation is therapist</td>
</tr>
<tr>
<td>userOcc = student</td>
<td>User occupation is student</td>
</tr>
<tr>
<td>option = OOOS</td>
<td>The button that user click is “Connect to one-on-one therapy session”</td>
</tr>
<tr>
<td>option = ER</td>
<td>The button that user click is “External resources”</td>
</tr>
<tr>
<td>entry/W.capacity +=1</td>
<td>The entry action of the waiting room. Every time when the waiting room state is entered, the capacity of the waiting room will increase by 1</td>
</tr>
<tr>
<td>exit/W.capacity -=1</td>
<td>The exit action of the waiting room. Every time when the waiting room state is exited, the capacity of the waiting room will decrease by 1.</td>
</tr>
<tr>
<td>when(isFirst = username &amp;&amp; M.capacity &lt;= 1 &amp;&amp; M.Therapist &lt;= 1 &amp;&amp; M.student = 0)</td>
<td>Only when a student is first in waiting room, and the number of people in meeting room is less than or equal to 1, and the number of therapist in the meeting room is less or equal to 1, and there is no student in the meeting room, then the student in the first place in the waiting room can enter the meeting room.</td>
</tr>
<tr>
<td>entry /M.capacity +=1</td>
<td>The entry action of the meeting room. Every time when the meeting room state is entered, the capacity of the meeting room will increase by 1</td>
</tr>
<tr>
<td>entry [userOcc = therapist]/M.therapist += 1</td>
<td>If a therapist enter the meeting room, the number of therapists in the meeting room will increase by 1</td>
</tr>
<tr>
<td>entry [userOcc = student]/M.student += 1</td>
<td>If a student enter the meeting room, the number of students in the meeting room will increase by 1</td>
</tr>
<tr>
<td>exit /M.capacity -=1</td>
<td>The exit action of the meeting room. Every time when the meeting room state is exited, the capacity of the meeting room will decrease by 1.</td>
</tr>
<tr>
<td>exit [userOcc = therapist] /M.therapist -= 1</td>
<td>If a therapist exit the meeting room, the number of therapists in the meeting room will decrease by 1</td>
</tr>
<tr>
<td>exit [userOcc = student] /M.student -= 1</td>
<td>If a student exit the meeting room, the number of students in the meeting room will decrease by 1</td>
</tr>
</tbody>
</table>
3.3 Quality Requirements

We perform quality requirements interviews by applying the 100-dollar prioritization chart, and obtain the top three quality requirements of our application.

They are usability, performance and security, and below is the result of the 100-dollar prioritization chart:

<table>
<thead>
<tr>
<th>Scores</th>
<th>Requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Scalability</td>
</tr>
<tr>
<td>Student 1</td>
<td>5</td>
</tr>
<tr>
<td>Student 2</td>
<td>10</td>
</tr>
<tr>
<td>Student 3</td>
<td>15</td>
</tr>
<tr>
<td>Student 4</td>
<td>10</td>
</tr>
<tr>
<td>Student 5</td>
<td>10</td>
</tr>
<tr>
<td>Therapist 1</td>
<td>5</td>
</tr>
<tr>
<td>Sum</td>
<td>55</td>
</tr>
</tbody>
</table>

Below is a description of each requirement and how we plan on measuring them:

- **Usability**: Users should connect to a therapist easily, for example, within a few clicks. The app should be self-explanatory without too steep of a learning curve.
  
  - Measurements: time spent to send in a request to a therapist.

- **Performance**: The matching time between a student and a therapist should be within an hour if the students attempt to connect on a regular workday between 9 am-5 pm. If a student is trying to connect outside that time frame, they should expect to receive a response by the next business day.

  - Measurements: average waiting time (mins) of the matching time on a regular workday between 9 am-5 pm and outside this time frame.

- **Security**: Users’ data are protected. Only therapists and the mental health office have access to transcripts and notes from each therapy session.

  - Measurements: percentage of successful attacks.

Below is the rich fit criteria table elicitated from the quality requirement interview:

<table>
<thead>
<tr>
<th>Outcomes</th>
<th>Outstanding</th>
<th>Target</th>
<th>Minimum</th>
</tr>
</thead>
<tbody>
<tr>
<td>Usability (time spent to send in a request to a therapist)</td>
<td>1.67 mins</td>
<td>6.33 mins</td>
<td>15.83 mins</td>
</tr>
<tr>
<td>Performance (average waiting time (mins) of the matching time on a regular workday between 9 am-5 pm and outside this time frame.)</td>
<td>5.17 mins</td>
<td>19.17 mins</td>
<td>45.83 mins</td>
</tr>
<tr>
<td>Security (percentage of successful attacks.)</td>
<td>0.33%</td>
<td>4.5%</td>
<td>9.17%</td>
</tr>
</tbody>
</table>
The rich fit criteria table summarizes the outstanding, target, and minimum criteria of our top three quality requirements. It contains the mean and standard deviation, and the top three use cases is determined by tallying up all the results from the 100 dollar prioritization exercise.

For the rich fit criteria, our team performed several quality requirement interviews by asking the interviewees their outstanding, target, and minimum criteria for each of the three requirements, then we calculate the mean and Std to achieve the final result. However, the outstanding criterias are unreliable because of its mean and Std value, so we are aiming to achieve the target and minimum criteria for each of the requirements.
Index

Requirements, 2