Ryerson University

Department of Electrical, Computer, & Biomedical Engineering
Faculty of Engineering & Architectural Science

Course Title:	Software Engineering
Course Number:	CPS888
Semester/Year (e.g. F2017)	W2021

Instructor	Dr. Sadaf Mustafiz

Assignment/Lab Number:	3
Assignment/Lab Title:	Phase 2 Report

Submission Date:	March 22, 2021
Due Date:	March 22, 2021

Student LAST Name	Student FIRST Name	Student Number	Section	Signature*
Ho	Brandon	500727531	01	BH
Chaudhari	Soumya	500758675	01	SC
Puvanenthira	Nirahulan	500823110	01	NP
Shreekant	Vatsal	500771363	01	VS

*By signing above you attest that you have contributed to this written lab report and confirm that all work you have contributed to this lab report is your own work. Any suspicion of copying or plagiarism in this work will result in an investigation of Academic Misconduct and may result in a "0" on the work, an "F" in the course, or possibly more severe penalties, as well as a Disciplinary Notice on your academic record under the Student Code of Academic Conduct, which can be found online at: http://www.ryerson.ca/senate/current/pol60.pdf

Team Information

Table 1: Team #2

Name	Student ID	Email Address
Soumya Chaudhari	500758675	s28chaud@ryerson.ca
Brandon Ho	500727531	b1ho@ryerson.ca
Nirahulan Puvanenthira	500823110	npuvanenthira@ryerson.ca
Vatsal Shreekant	500771363	vatsal.shreekant@ryerson.ca

1.1 Task 1- Architectural Design

The architectural design pattern that we chose to use was a combination of Client and Server with MVC(Model View Controller). We chose a client and server to differentiate between the frontend and the backend respectively. The client will send a request to the server from the browser and the server will respond back with a response using the browser. This means that client and server use the browser to communicate to one-another. The benefit of using Client and Server is that it is ideal for modelling a set of services where clients can request them which fits our project as the client should be able to request a slew of services[1]. The cons of this architectural design is that requests are typically handled in separate threads on the server and inter-process communication requires overhead[1]. To mitigate the backend(i.e. server) issues, we decided to make the backend a MVC and this is depicted in detail in Figure 1. For the frontend (i.e. Client), we know it will be used by different users such as Student, Employee, HR Manager, Counsellor , Therapist, Company, School and Sentiment Admin.

The MVC architecture in the backend is divided into three main logical components of Model, a View and Controller. The model contains the core functionality and data, the view displays the information to the user (more than one view may be defined) and the controller handles the input from the user[1]. The advantage of using the MVC pattern is that it is easy to have multiple views of the same model and the views can be connected or disconnected at run-time[1]. Some other advantages of MVC include faster development, easy for multiple developers to collaborate and work together, easy to update the application and easy to Debug as we have multiple levels properly written in the application[2]. The downside is that the model can become complex with time making it hard to understand[2]. As well, it must have strict rules on methods[1]. In our design pattern, the server is what harbours the Model, View and Controller. The controller manipulates data from the model. Whereas, the model will update and notify change to the view. The Model also has the Database server.

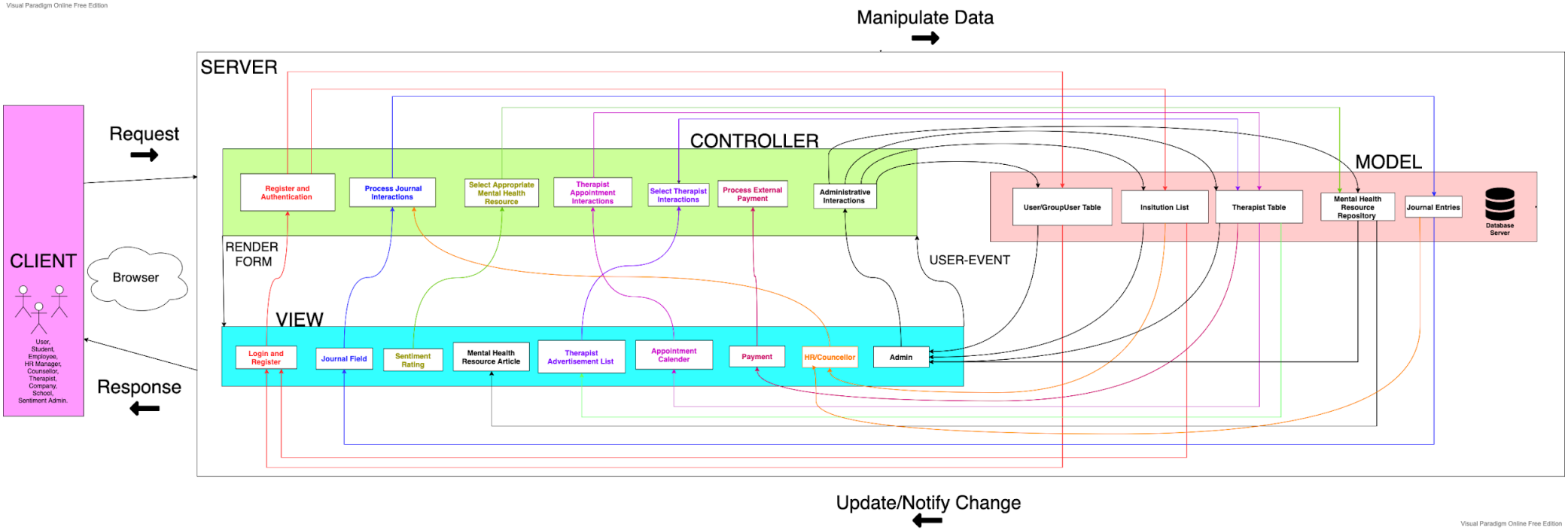

Figure 1 : Architectural Diagram (MVC + Server Client Model)

1.2 Task 2 - Detailed Design

- Task 2.1: Interaction Model (Sequence Diagrams)

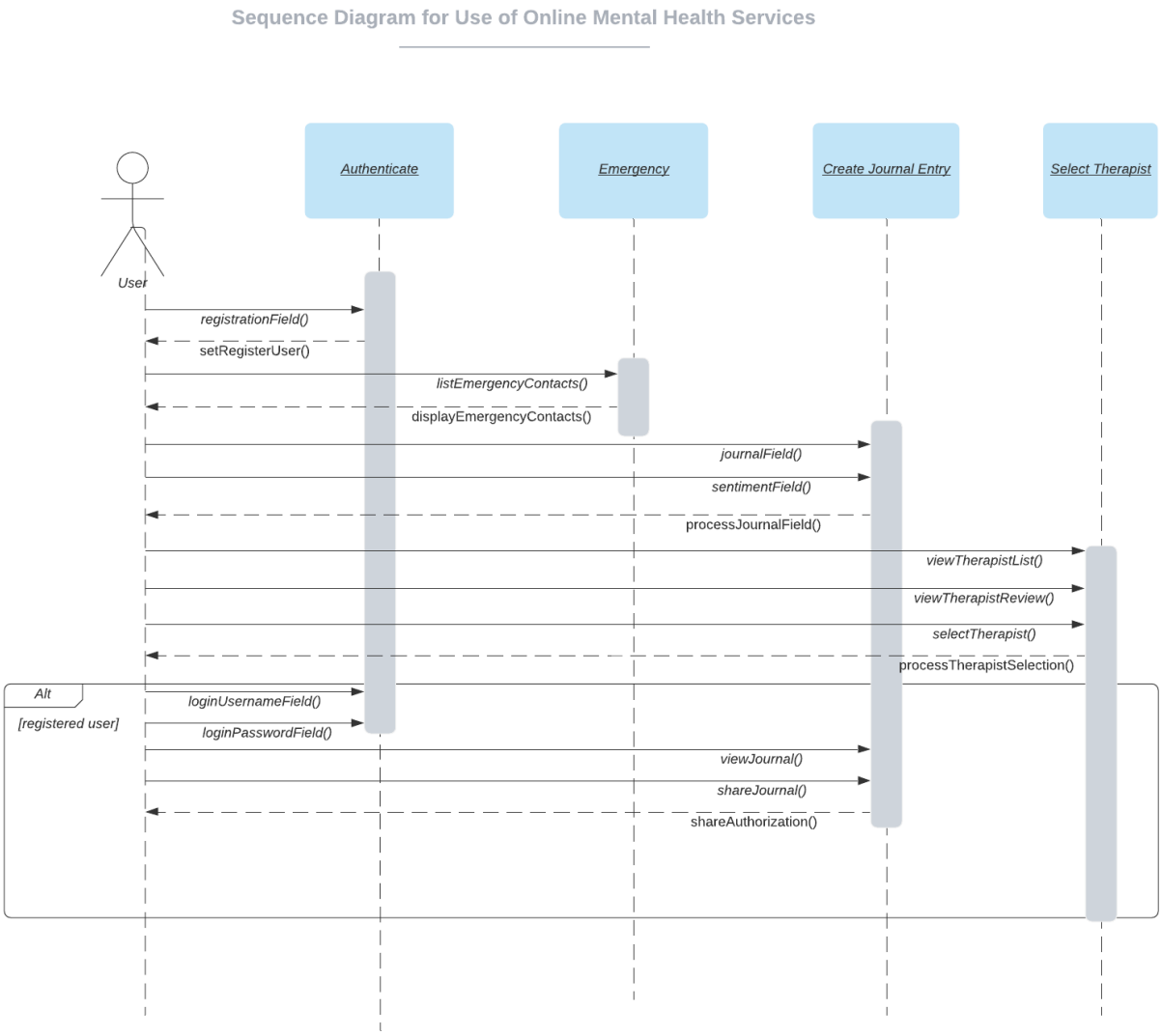

Figure 2: Sequence Diagram for Use of Online Mental Health Resources Use Case

The above figure shows how the User (actor) interacts within the system to access the Use of Online Mental Health Services.

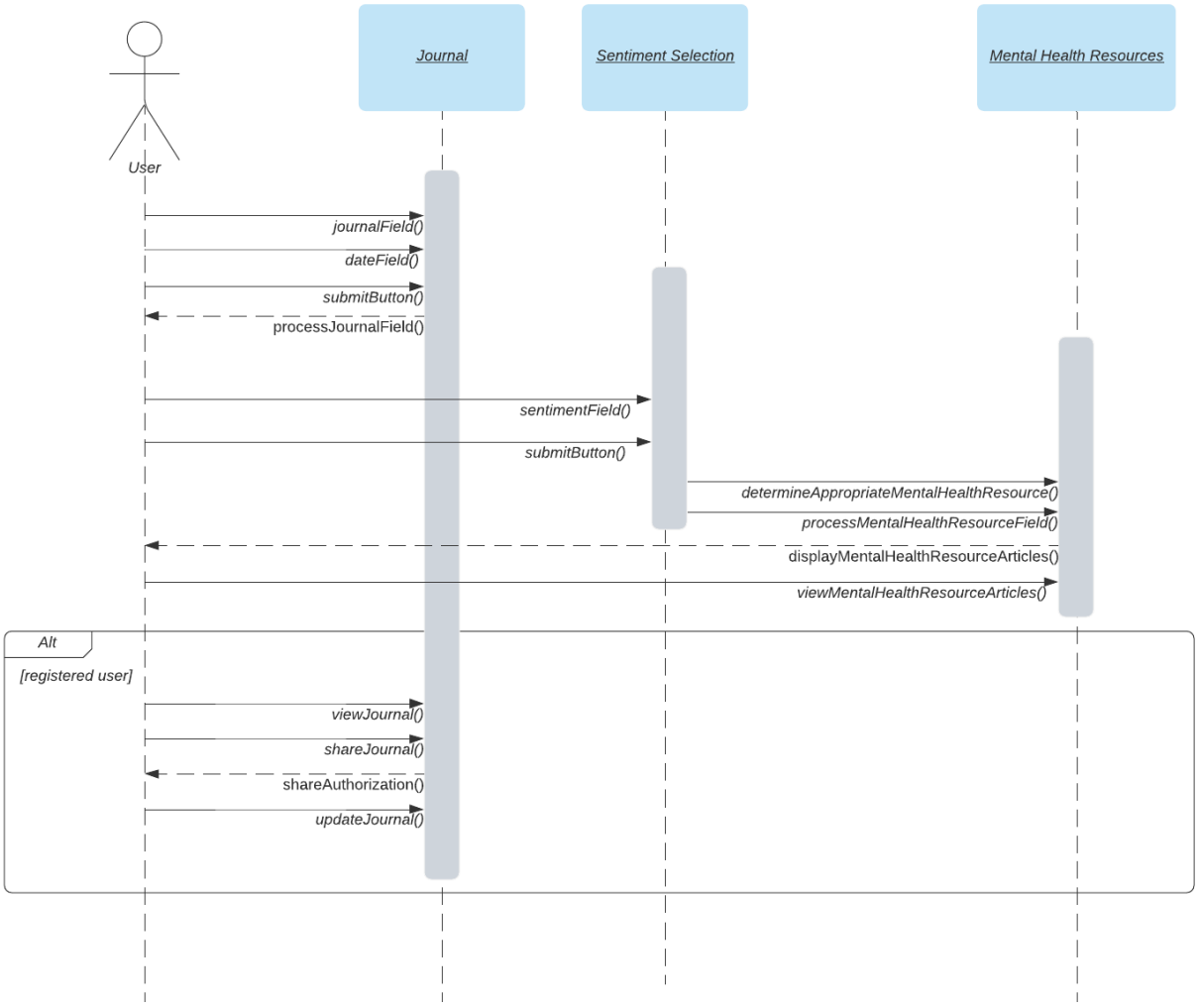

Figure 3: Sequence Diagram for Create Journal Entry Use Case

The above figure shows how the User (actor) interacts within the system to successfully create a journal entry.

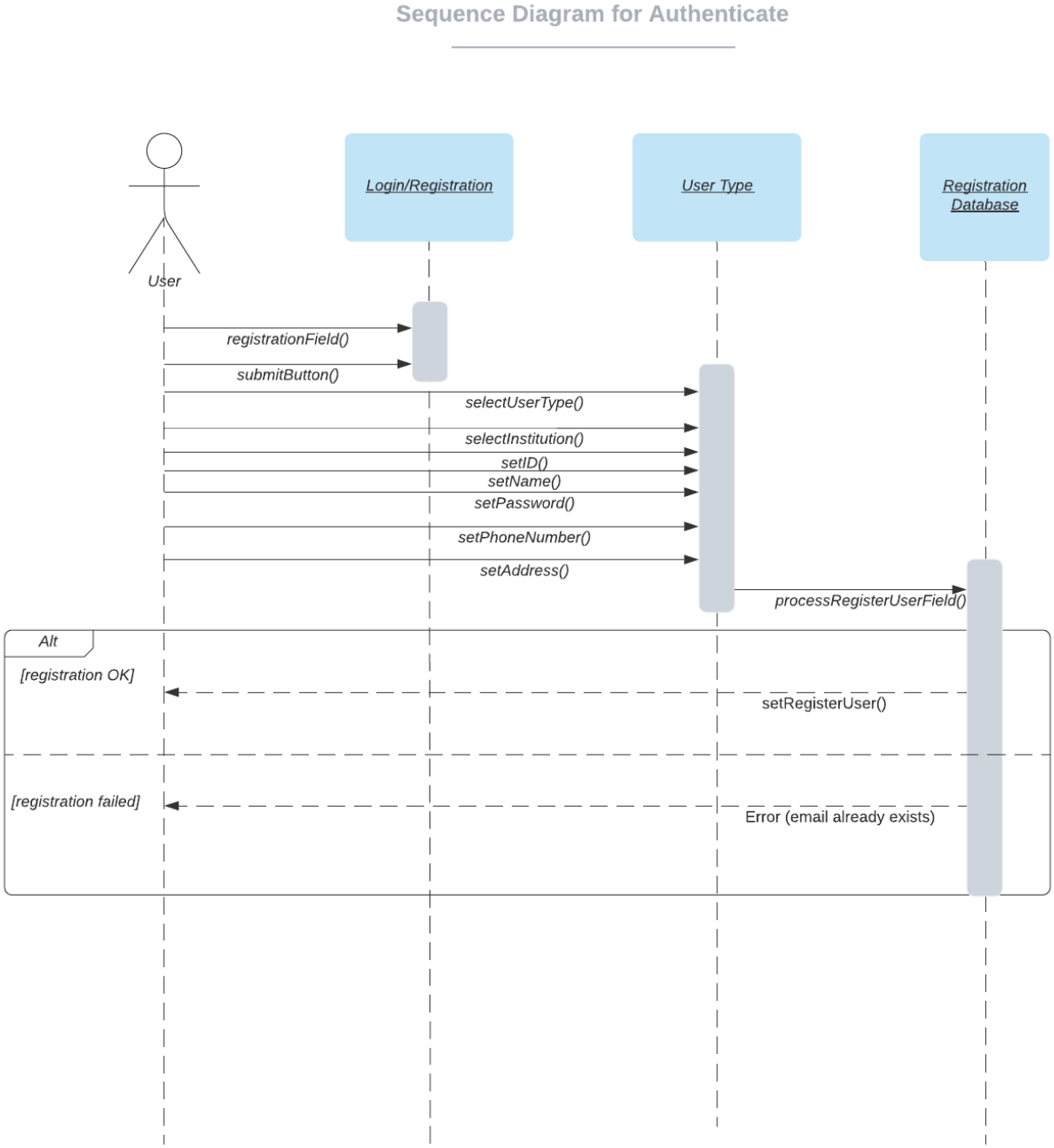

Figure 4: Sequence Diagram for Authenticate Use Case

The above figure shows how the User (actor) interacts within the system to successfully authenticate the login/registration process.

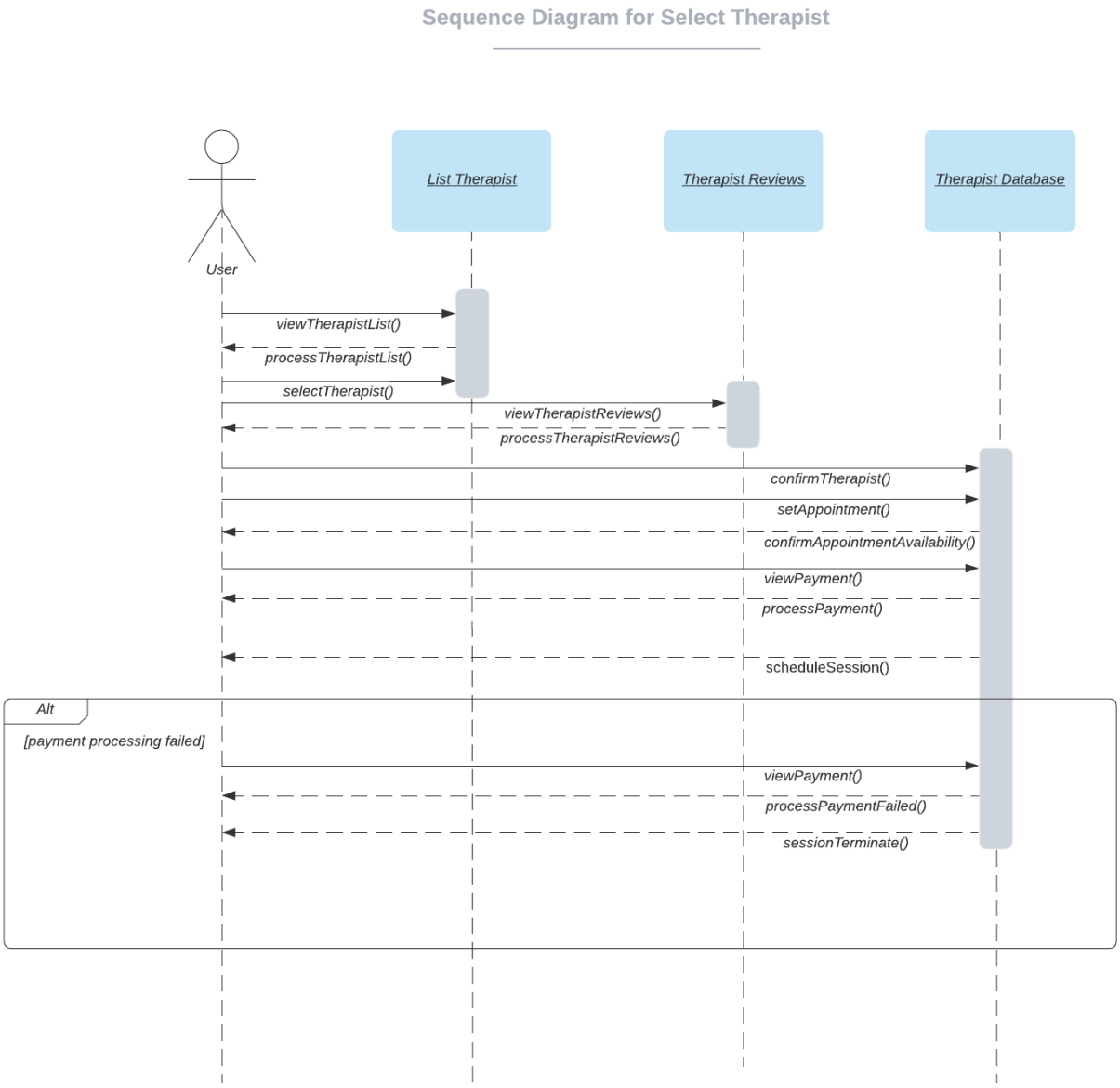

Figure 5: Sequence Diagram for Select Therapist Use Case

The above figure shows how the User (actor) interacts within the system to select a Therapist, process a payment and successfully schedule an appointment.

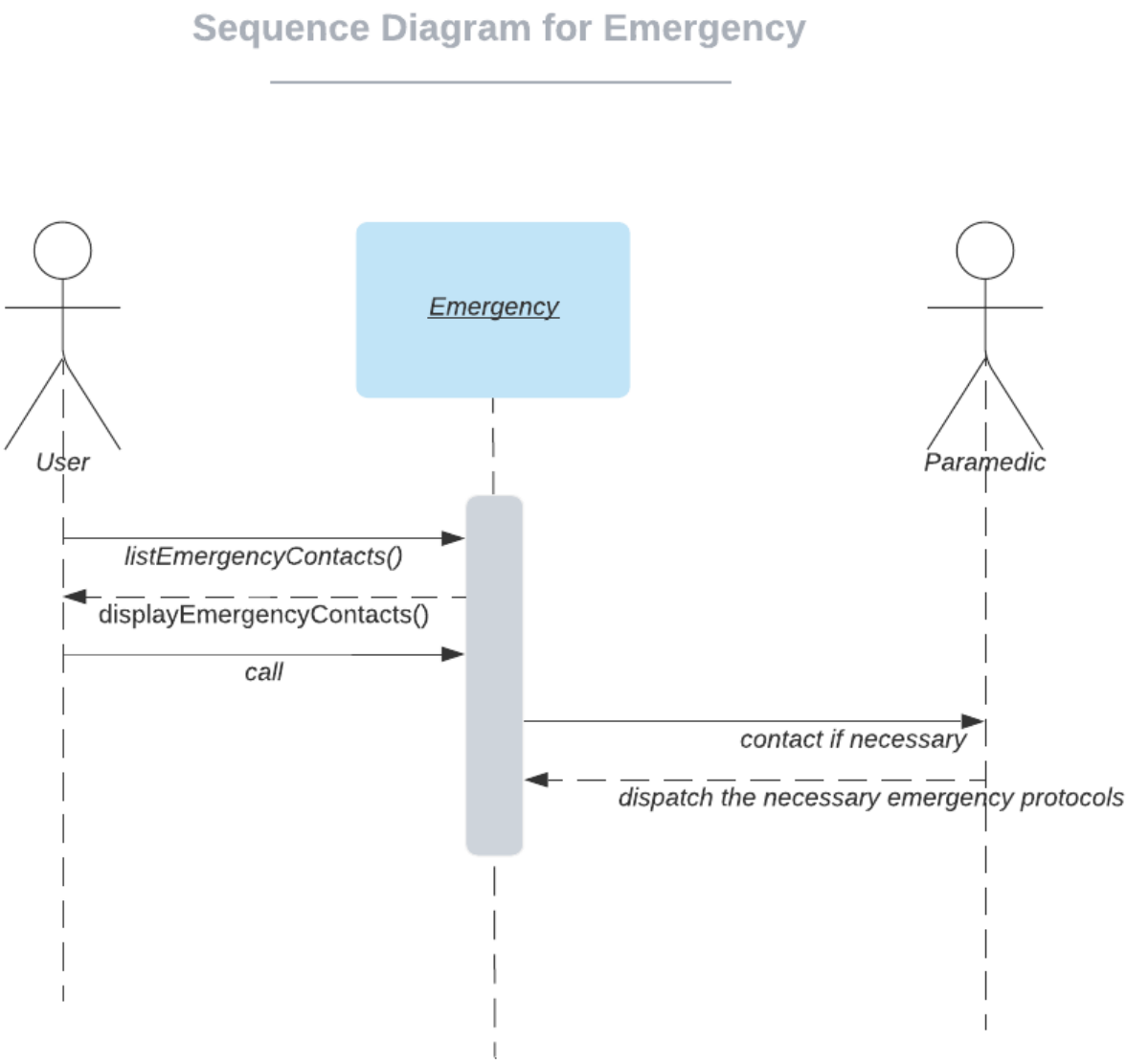

Figure 6: Sequence Diagram for Emergency Use Case

The above figure shows the interaction between a User (actor) and a Paramedic (actor) within the system in the case of an emergency.

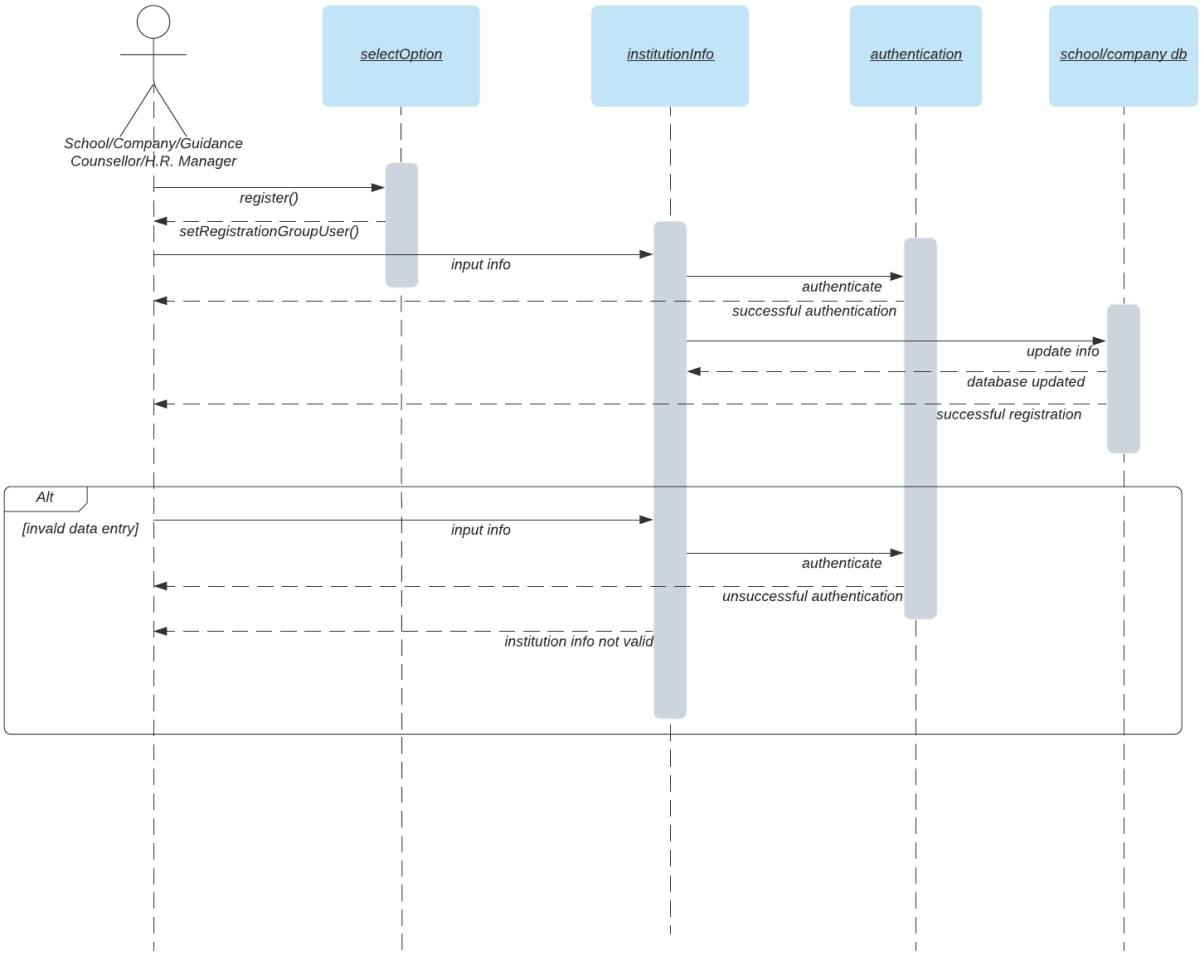

Figure 7: Sequence Diagram for Group Register Use Case

The above figure shows how the School/Company/Guidance Counsellor/H.R. Manager (actor) interacts within the system to form a group register.

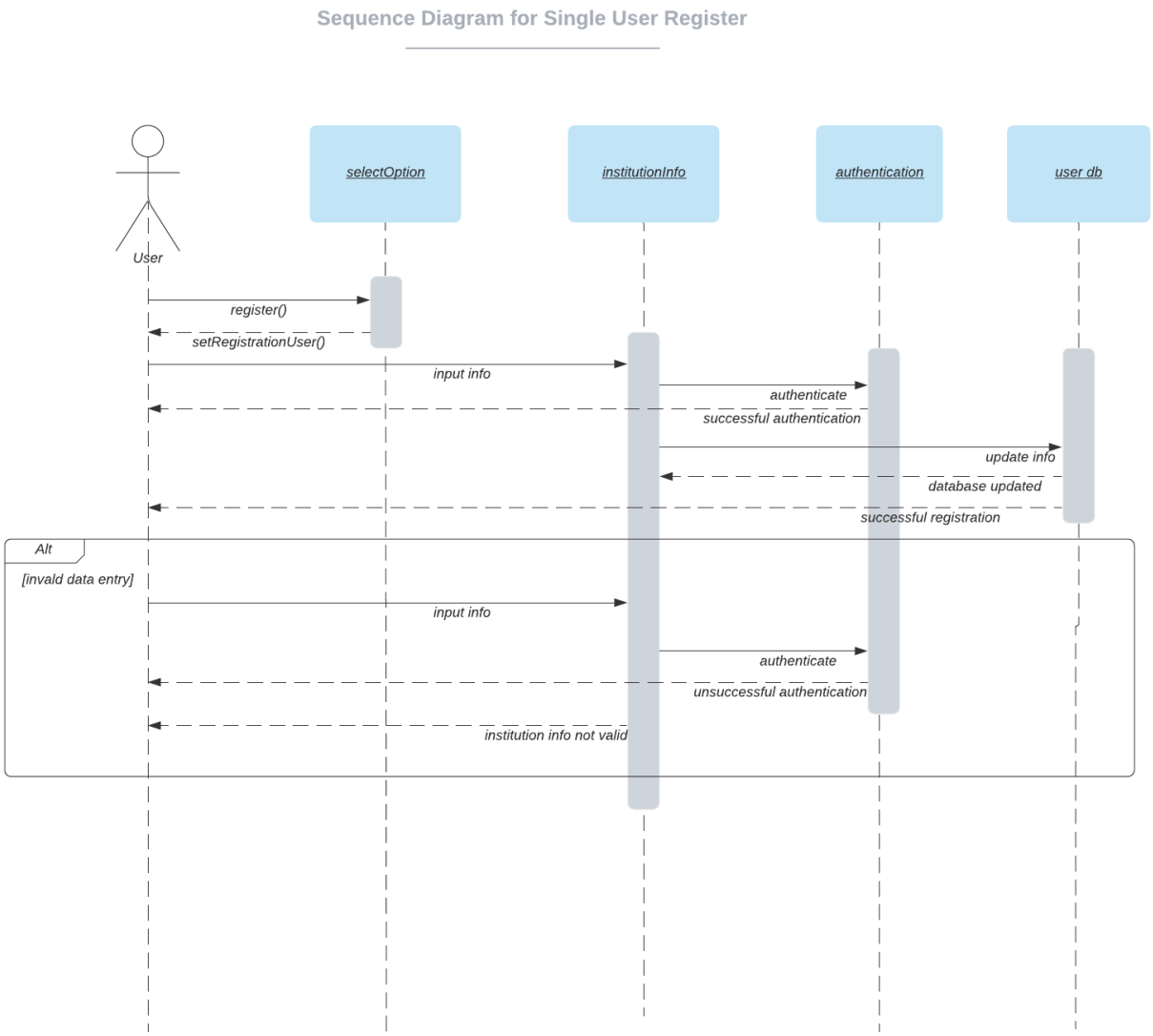

Figure 8: Sequence Diagram for Single User Register Use Case

The above figure shows how the User (actor) interacts within the system to register as a stand alone user.

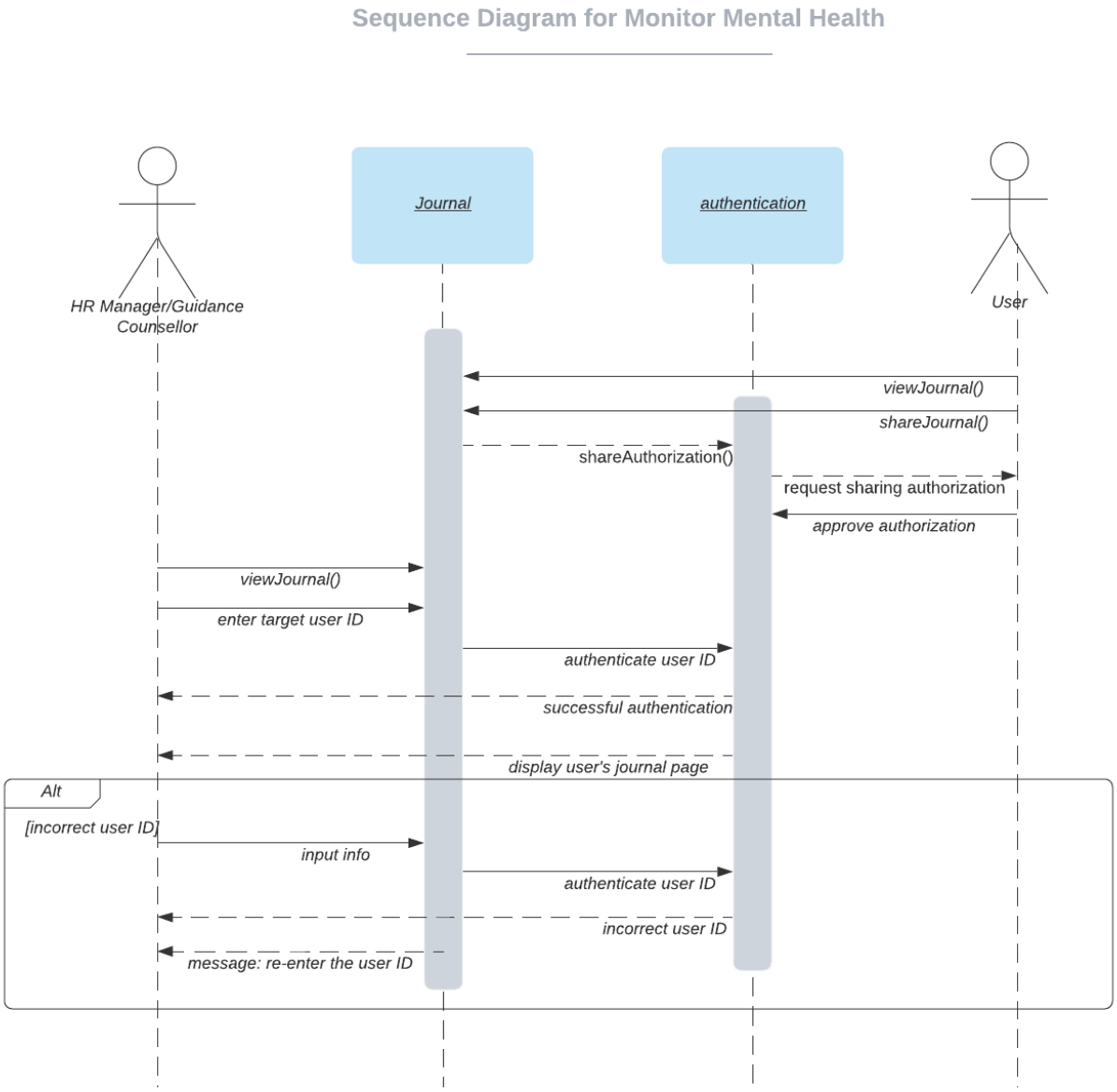

Figure 9: Sequence Diagram for Monitor Mental Health Use Case

The above figure shows how the H.R. Manager/Guidance Counsellor (actor) interacts within the system to view the target user's (actor) journal entries.

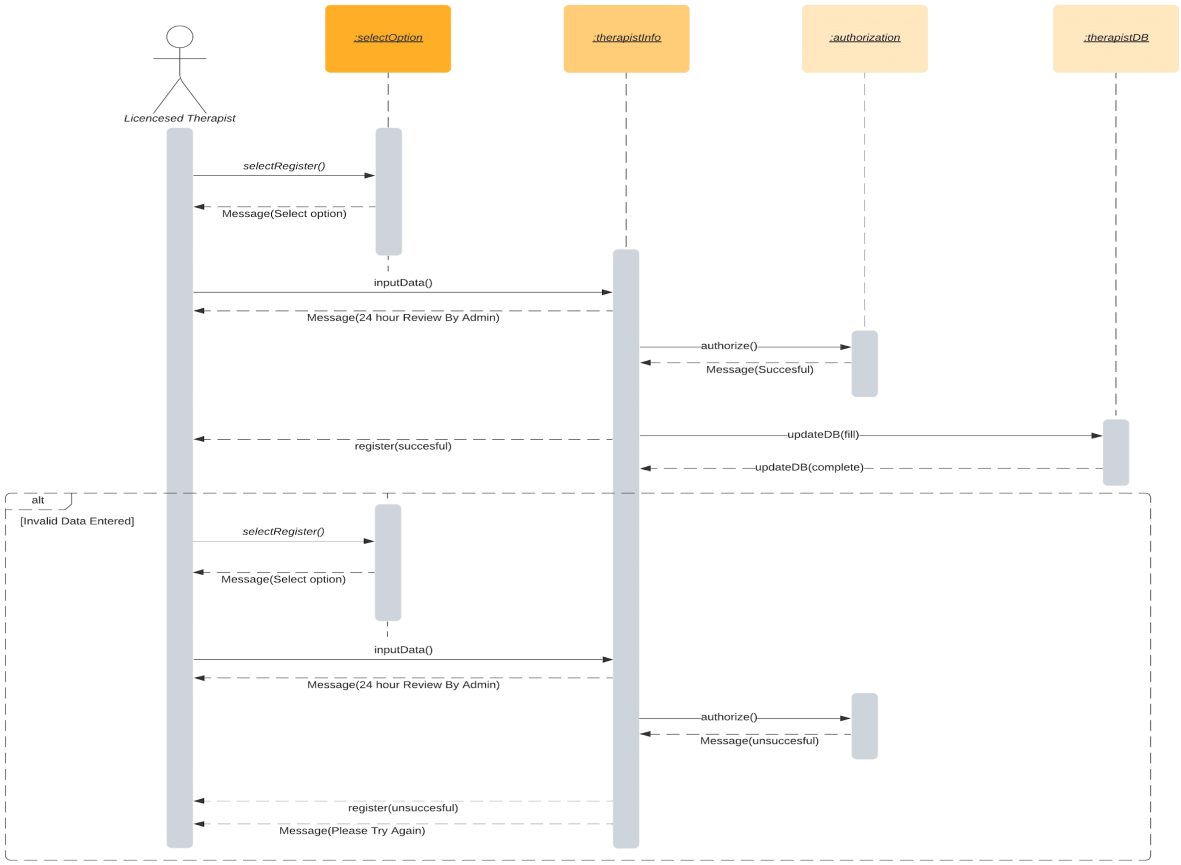

Figure 10: Sequence Diagram for Therapist Register Use Case

The above figure shows how a Therapist interacts with the system to self register.

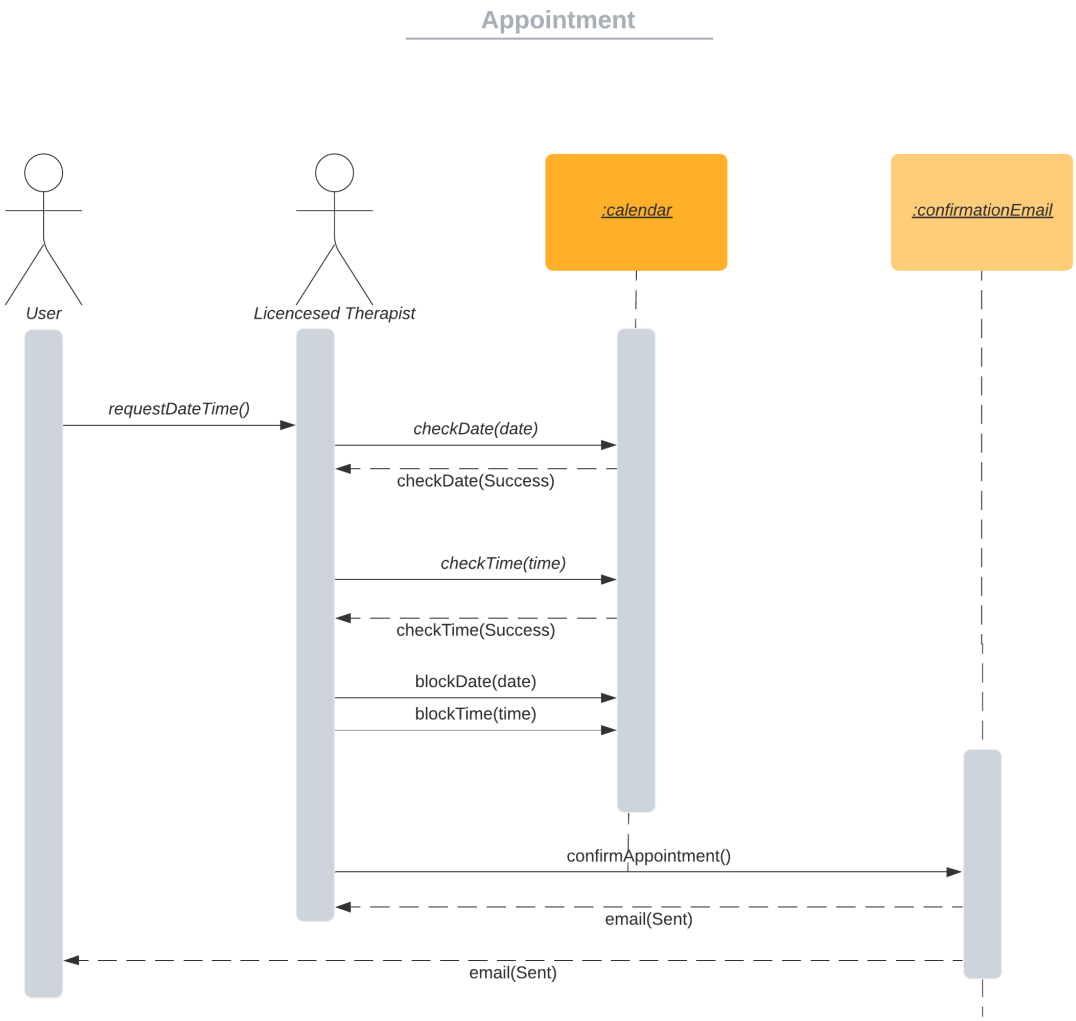

Figure 11: Sequence Diagram for Appointment Use Case

The above figure shows how the Licensed Therapist (actor) interacts within the system to book an appointment created by the User (actor).

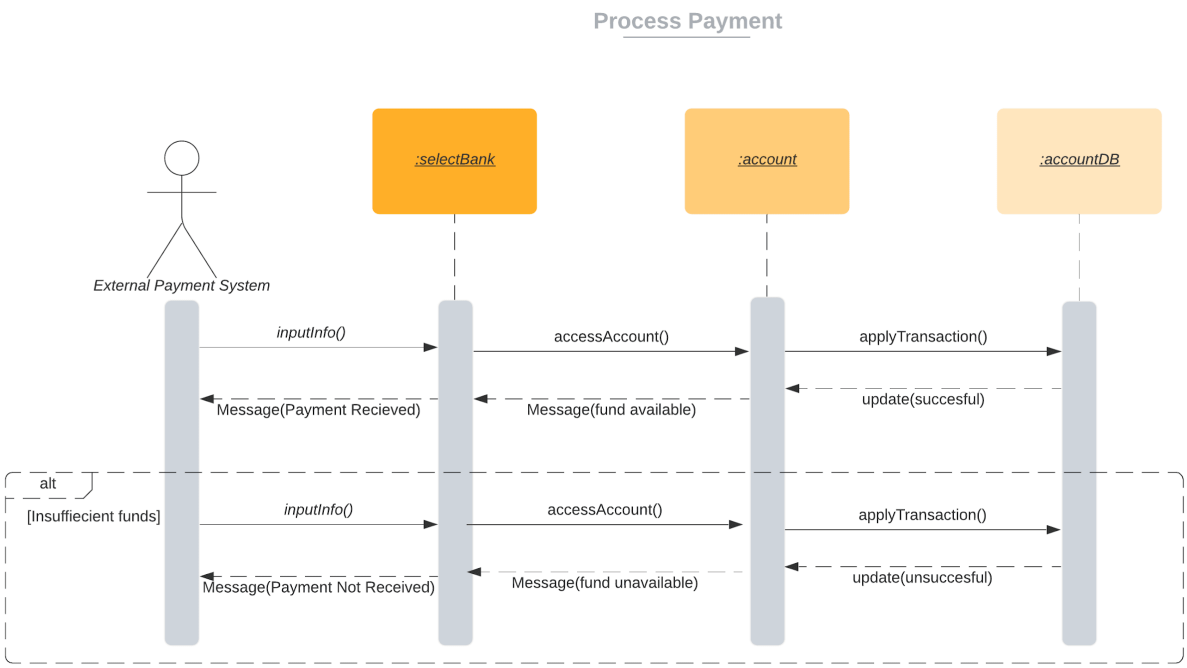

Figure 12: Sequence Diagram for Process Payment Use Case

The above figure shows how the External Payment System (actor) interacts within the system to successfully process a payment.

- Task 2.2 Design Class Model

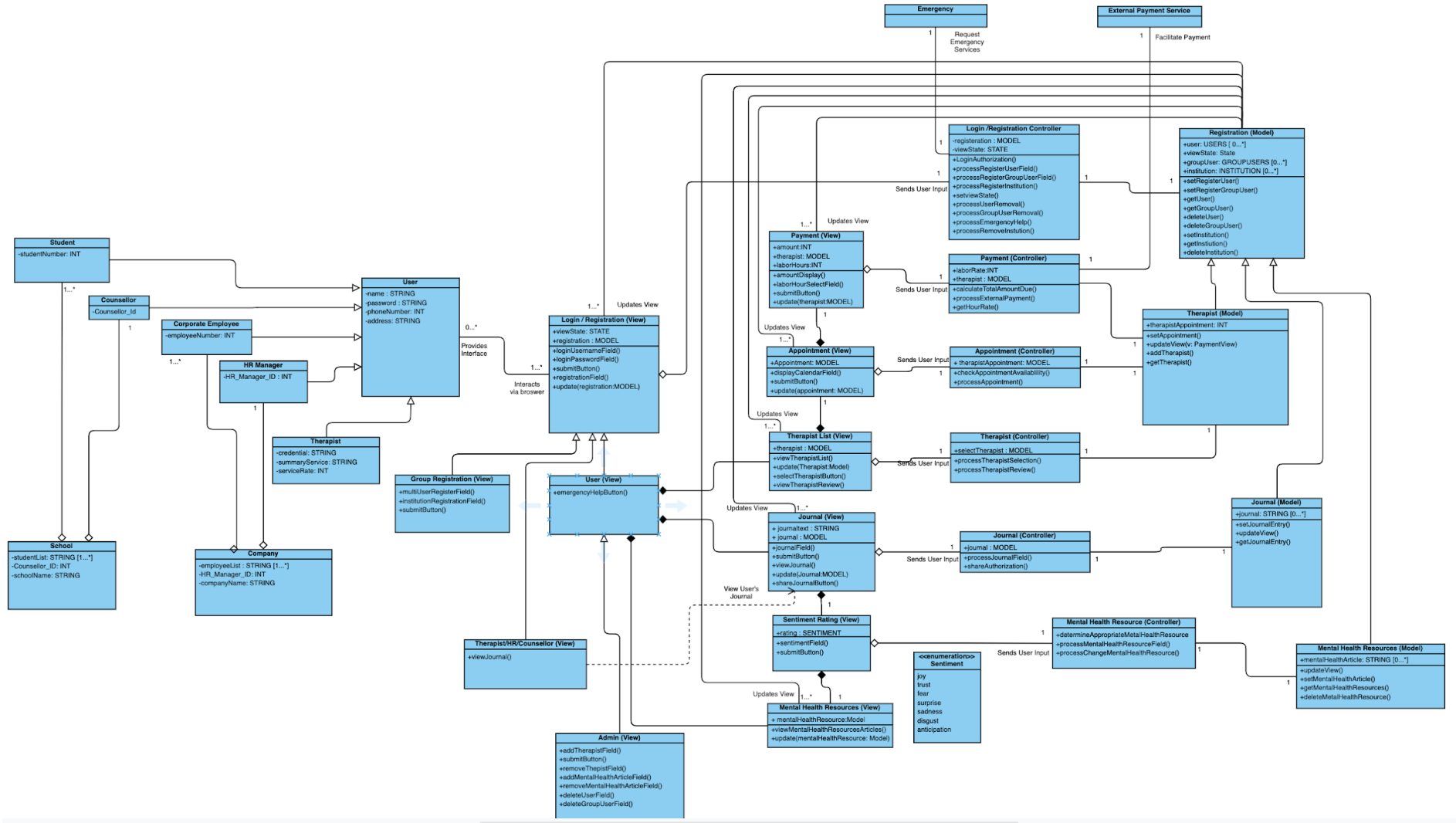

Figure 13: Design Class Diagram (MVC)

- Task 2.3: State Machine Model

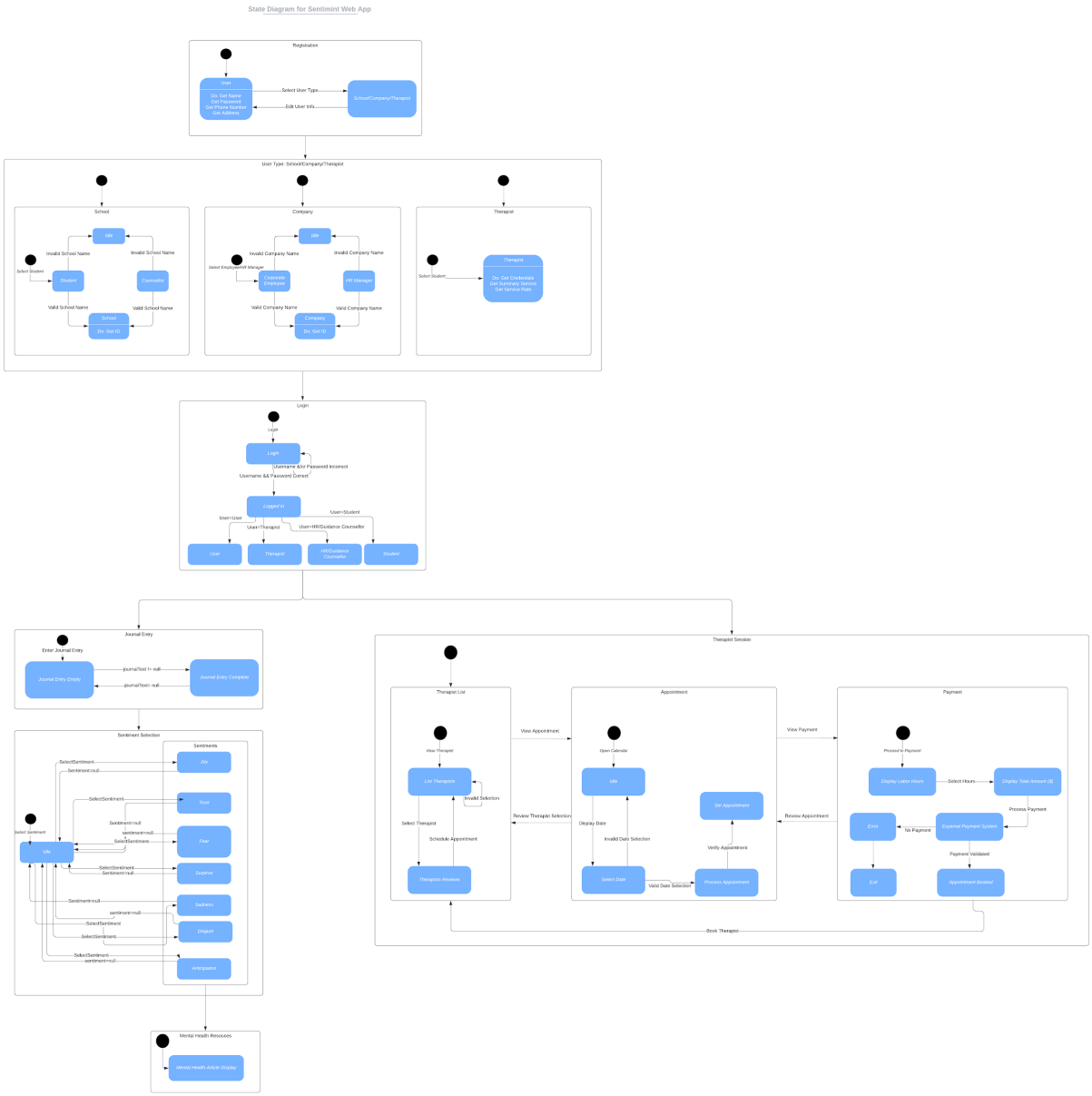

Figure 14: State Machine Diagram for Sentimint Web Application

1.3 Task 3 - Test Design

Authenticate Test Case

The User authenticates using the login portal.

Table 2.1: Authenticate Test Case: A

Test cases	Steps	Input	Expected Result
A	Authenticate		
1	User Enter account details	….	User can enter the specific details
2	System Informs user of successful login	….	User views success message

Table 2.2: Authenticate Test Case: B

Test cases	Steps	Input	Expected Result
B	Authenticate-Invalid Information		
1	Enter account details	….	User can enter (incorrect) details
2	System Informs user of unsuccessful login	….	User views error message

Access to Self-help Resources Test Case

User access Mental Health Resources for self improvement.

Table 3: Access to Self-Help Resources Test Case: A

Test cases	Steps	Input	Expected Result
A	Access to Self-help Resources		
1	User views list of self-help resources		User views list of self-help resources
2	User can pick from a list of self-help resources.		User selects which self-help resource would help them the most.

Select Therapists Test Case

User procures service of therapist from “Therapist Advertisement List”.

Table 4.1: Select Therapist Test Case: A

Test cases	Steps	Input	Expected Result
A	Select Therapists		
1	User views list of therapists		User views list of therapists
2	User picks from therapist and pays the amount owing		User views success message

Table 4.2: Select Therapist Test Case: B

Test cases	Steps	Input	Expected Result
B	Select Therapists-Invalid Information		
1	User views list of therapists		User views list of therapists
2	System informs user of unsuccessful selection of therapist due to incorrect account information.		User views error message

Book Appointment Test Case

User books an appointment with a selected therapist.

Table 5: Book Appointment Test Case: A

Test cases	Steps	Input	Expected Result
A	Book Appointment		
1	User views available times with therapist	….	User views times of therapists available
2	User chooses a time that best suits their needs	….	User views success message

Group Register Test Case

An institution such as a school or company registers their students or employees to provide mental health support for their respective members.

Table 6: Group Register Test Case: A

Test cases	Steps	Input	Expected Result
A	Group Register		
1	Company/School picks group register option.	…	Company/School picks group register option.
2	Company/School is allowed to register their organization.	...	Company/School views successful message after group register option.

Single User Register Test Case

New user registers with Sentiment Mental Health web application.

Table 7.1: Single User Register Test Case: A

Test Case	Steps	Input	Expected Result
A	Single User Register		
1	User selects register button	…	View Single User Register, Group User or Therapist Register options
2	User selects single user select button	…	View data fields to enter name, password, location and phone number
3	User inputs user information into appropriate data fields	name, password, location, phone number	View successful registration message

Table 7.2: Single User Register Test Case: B

Test Case	Steps	Input	Expected Result
B	Single User Register (invalid/incomplete)		
1	User selects register button	…	View Single User Register, Group User or Therapist Register options
2	User selects single user select button	…	View data fields to enter name, password, location and phone number
3	User inputs user information into appropriate data fields and select submit button (Invalid/incomplete)	name, password, location, phone number (Invalid/incomplete)	(Invalid/incomplete) View unsuccessful registration message

Therapist Register Test Case

New Therapist registers services with Sentiment Mental Health web application.

Table 8.1: Therapist Register Test Case: A

Test Case	Steps	Input	Expected Result
A	Therapist Register		
1	User selects register button	…	View Single User Register, Group User or Therapist Register options
2	User selects Therapist select button	…	View data fields to enter name, password, credentials, phone number, location and summary of services

3	User inputs user information into appropriate data fields	name, password, credentials, phone number, location and summary of services	View successful registration and 24hour review notice
4	Admin reviews therapist application	…	Admin can view Therapist application
5	Admin manually approves Therapist by selecting therapist and selecting add button	…	Therapist added to “Therapist Advertisement List”

Table 8.2: Therapist Register Test Case: B

Test Case	Steps	Input	Expected Result
B	Therapist Register (invalid/incomplete)		
1	User selects register button	…	View Single User Register, Group User or Therapist Register options
2	User selects Therapist user select button	…	View data fields to enter name, password, credentials, phone number, location and summary of services
3	User inputs user information into appropriate data fields (Invalid/incomplete)	name, password, credentials, phone number, location and summary of services (Invalid/incomplete)	(Invalid/incomplete) View unsuccessful registration message

Table 8.3: Therapist Register Test Case: C

Test Case	Steps	Input	Expected Result
C	Therapist Register (not approved)		
1	User selects register button	…	View Single User Register, Group User or Therapist Register options
2	User selects Therapist user select button	…	View data fields to enter name, password, credentials, phone number, location and summary of services
3	User inputs user information into appropriate data fields	name, password, credentials, phone number, location and summary of services	View successful registration and 24hour review notice
4	Admin reviews therapist application	…	Admin can view Therapist application
5	Admin manually disapproves Therapist by selecting therapist and selecting remove button	…	Therapist removed from Admin application Therapist not added to “Therapist Advertisement List”

Process Payment Test Case

User makes payment to Therapist for services.

Table 9.1: Process Payment Test Case: A

Test Case	Steps	Input	Expected Result
A	Process Payment		
1	User presented with total amount due User pays Therapist through an external payment service.	…	Therapist received confirmation of payment by external payment service. Appointment confirmation message for both User and Therapist. Appointment added to database.

Table 9.2: Process Payment Test Case: B

Test Case	Steps	Input	Expected Result
B	Process Payment (no/invalid payment)		
1	User presented with total amount due User does not pay Therapist or pays an incorrect amount to the external payment service.	…	Error message presented for no confirmation of payment to User and Therapist.

Create Journal Entry Test Case

User creates a journal entry and selects the most appropriate sentiment to describe their current state. Consequently, the user receives the most appropriate “Mental Health Resources Articles” based on sentiment selected.

Table 10: Create Journal Entry Test Case: A

Test Case	Steps	Input	Expected Result
A	Create Journal Entry		
1	User input “thoughts/sentiments” into journal field and selects submit button	Text “thought/sentiment”	Journal entry recorded to database
2	User selects the Sentiment that best describes their state from the list.	...	Presented appropriate Mental Health Resources

Monitor Mental Health Test Case

HR manager or school guidance counsellor reviews the mental state of associated students or employees based on contents of shared journals entries.

Table 11: Monitor Mental Health Test Case: A

Test Case	Steps	Input	Expected Result
A	Monitor student/employee journal		
1	HR/Counsellor selects student/employee of interest from list who have shared journal entries and select submit button	…	View student/employee journal

Emergency Test Case

In case of emergency, users can request help via external emergency services.

Table 12: Emergency Test Case: A

Test Case	Steps	Input	Expected Result
A	User request emergency help		
1	User selects emergency help button	…	Emergency request sent to most appropriate services based on location of user.

REFERENCES:

1. Mallawaarachchi, V. (2020, September 02). 10 common software architectural patterns in a nutshell. Retrieved March 5, 2021, from https://towardsdatascience.com/10-common-software-architectural-patterns-in-a-nutshell-a0b47a1e9013
2. What is mvc? Advantages and disadvantages of mvc. (2019, October 22). Retrieved March 10, 2021, from https://www.interserver.net/tips/kb/mvc-advantages-disadvantages-mvc/