

Software Engineering

# WEB-CBT

Patryk Prusak, Rana Sahin, Akshma Atreja, Mikołaj Stańczyk, Sujith Madesh, Mina Hany

**Abstract.** Proposition of a system consisting of a web app, server and a database that allows users to apply a fairly new type of therapy TEAM-CBT to provide relief of depression, anxiety and symptoms of other mental issues.

## KEY WORDS

1. TEAM-CBT
2. Mental Health
3. Therapy

# Contents

<b>1 Introduction</b>	<b>3</b>
1.1 State Of Mental Health . . . . .	3
1.2 New Approach To Recovery . . . . .	3
1.3 Neuroscience Of Change . . . . .	8
1.4 Implementation of TEAM-CBT . . . . .	10
<b>2 System Requirements</b>	<b>11</b>
2.1 User Stories . . . . .	11
2.2 Critical Requirements . . . . .	16
2.3 Use Cases . . . . .	22
<b>3 Architecture Design</b>	<b>25</b>
<b>4 System Objects</b>	<b>26</b>
<b>5 System Structure</b>	<b>27</b>
<b>6 System States</b>	<b>30</b>
<b>7 System Activity</b>	<b>37</b>
<b>8 Communication Protocol</b>	<b>44</b>
8.1 General Design . . . . .	44
8.2 Sequence Diagrams . . . . .	44
8.3 API . . . . .	55
<b>9 User Simulations</b>	<b>73</b>
<b>10 Abnormal behaviour</b>	<b>76</b>
<b>11 Implementation Technology</b>	<b>77</b>
<b>12 Final Words</b>	<b>78</b>
<b>13 Glossary</b>	<b>78</b>
<b>14 Appendix A</b>	<b>79</b>
<b>15 Appendix B</b>	<b>81</b>
<b>16 Further Reading</b>	<b>86</b>
List of Tables	87
List of Figures	87

## 1. Introduction

*1.1. State Of Mental Health*—In recent years, there has been a large increase in the number of cases related to mental health problems. The current global pandemic situation has additionally increased the dynamics of the reported number of new cases of mental problems, both among adults, children and adolescents. The average share of adults reporting symptoms of anxiety disorder and/or depressive disorder has risen from 11% to 41.1% in a span of two years.\* One of five adults in the U.S experience mental illness each year,† one in six youth aged 6-17 in the U.S experience mental health disorder each year,‡ on top of that suicide is the second leading cause of death among people aged 10-34.§ These statistics are quite depressing on their own. In addition, mental health problems often lead to other chronic health and life-threatening conditions, to name a few: significantly higher risk of developing cardiovascular and metabolic diseases,¶ substance use disorder.|| In the social and sociological aspect, mental health problems often make normal functioning and life impossible, in young people they can, among others, increase the risk of dropping out of college or repeating a grade.||,§§ We all experience fear, stress and we all feel sad or depressed at times. There are, of course, situations in which these feelings are appropriate and helpful, but there are also situations when these feelings are undesirable and interfere with our daily lives, then therapy can be of great help to all of us, even when we don't qualify for a disorder according to the DSM-V.||| The presented data indicate that there is a strong need to change the approach to mental health treatment. To counteract these negative psychological changes, new types of therapy are being developed, and TEAM-CBT is a new way of conducting therapy, very effective in solving wide range of problems.\*

*1.2. New Approach To Recovery*—TEAM-CBT is an intervention consisting of four steps (in the case when this therapy is done on one's own the E step is omitted): T stands for Testing, E-Empathy, A-Paradoxical Agenda setting, M-Methods. The purpose of this project is to help the user complete these steps on their own. These steps take about two hours to complete, and the goal is to spend 10 to 30 minutes a day in the process. The user starts by describing the event that caused her/his upset, then marks what she/he feels in 10 different categories, evaluates the intensity (%) in each category, then writes down the thoughts that caused the mental state and identifies the Cognitive Distortions in each of them. That concludes the initial part of Testing. In Paradoxical Agenda Setting, the goal is

---

\* <https://www.kff.org/coronavirus-covid-19/issue-brief/the-implications-of-covid-19-for-mental-health-and-substance-use/>

† <https://www.samhsa.gov/data/sites/default/files/reports/rpt35325/NSDUHFRPDFWHTMLFiles2020/2020NSDUHFFR1PDFW102121.pdf>

‡ <https://jamanetwork.com/journals/jamapediatrics/fullarticle/2724377?guestAccessKey=f689aa19-31f1-481d-878a-6bf83844536a>

§ <https://www.nimh.nih.gov/health/statistics/suicide>

¶ <https://www.thelancet.com/commissions/physical-health-in-mental-illness>

|| <https://www.samhsa.gov/data/sites/default/files/reports/rpt35325/NSDUHFRPDFWHTMLFiles2020/2020NSDUHFFR1PDFW102121.pdf>

||| <https://www.nami.org/mhstats>

§§ <https://pubmed.ncbi.nlm.nih.gov/29195763/>

||| The Diagnostic and Statistical Manual of Mental Disorders - volume that defines and classifies mental disorders in order to improve diagnoses, treatment, and research

\* <https://www.feelinggoodinstitute.com/science-research-behind-approach>

to reduce the resistance to change that often slows down the recovery process. This is done by discovering that negative thoughts and feelings are not actually the result of a defect, but rather reflect what is great and valuable about us. This can be achieved by answering two questions for each of the category of feelings and for each thought:

1. What are some benefits, or advantages, of this negative thought or feeling? How might it be helping you?
2. What does this negative thought or feeling show about you and your core values that's positive and awesome?

After taking this step, the user may realize that getting rid of these negative feelings completely may not be beneficial, that it might be better to keep those unpleasant feelings, but at a lower intensity, thereby allowing their benefits to stay. This step ends with a desired % rating next to each feeling category. To move on to Methods, the user first need to select the most upsetting thought, which is the thought for which methods will be chosen. The methods are selected based on the distortions<sup>3845</sup> found in thought, but also based on the context of the situation. When the thought no longer feels upsetting, When the belief in a thought is reduced to 0 or close to 0, the user can move to another thought, possibly without choosing a new set of methods but using the one that already worked or as is often the case using the Externalization of Voices to defeat the rest of the thoughts. In Externalization of Voices the thoughts are defeated with the use of three approaches:

1. Acceptance Paradox
2. Self-Defense
3. Combination of the two above

After this process is done, it is important to rate the % in each of the feeling category. Finally, it can be checked whether the desired goals have been achieved and whether the session was helpful.

Why this approach works in short? It comes with the premise that we feel the way we think. Therefore, if we can change the way we think we can also change the way we feel. But there is more going on behind the scenes. All the beliefs and patterns of thought are reflected in the way our brains are wired. Changing these patterns leads to a physical change in the brain that leads to improvement in mood.<sup>†</sup> This process happens on paper by the use of following forms:

---

<sup>†</sup> <https://feelinggood.com/2019/11/18/167-feeling-great-professor-mark-noble-on-team-cbt-and-the-brain/>

**Daily Mood Journal\* Page 1 of 2**

Upsetting Event: \_\_\_\_\_

Emotions	% Now	% Goal	% After	Emotions	% Now	% Goal	% After
Sad, blue, depressed, down, unhappy				Embarrassed, foolish, humiliated, self-conscious			
Anxious, worried, panicky, nervous, frightened				Hopeless, discouraged, pessimistic, despairing			
Guilty, remorseful, bad, ashamed				Frustrated, stuck, thwarted, defeated			
Inferior, worthless, inadequate, defective, incompetent				Angry, mad, resentful, annoyed, irritated, upset, furious			
Lonely, unloved, unwanted, rejected, alone, abandoned				Other			

Negative Thoughts	% Now	% After	Distortions	Positive Thoughts	% Belief
1.					
2.					
3.					
4.					
5.					

Fig. 1. Daily Mood Journal

**Daily Mood Journal\* Page 2 of 2**

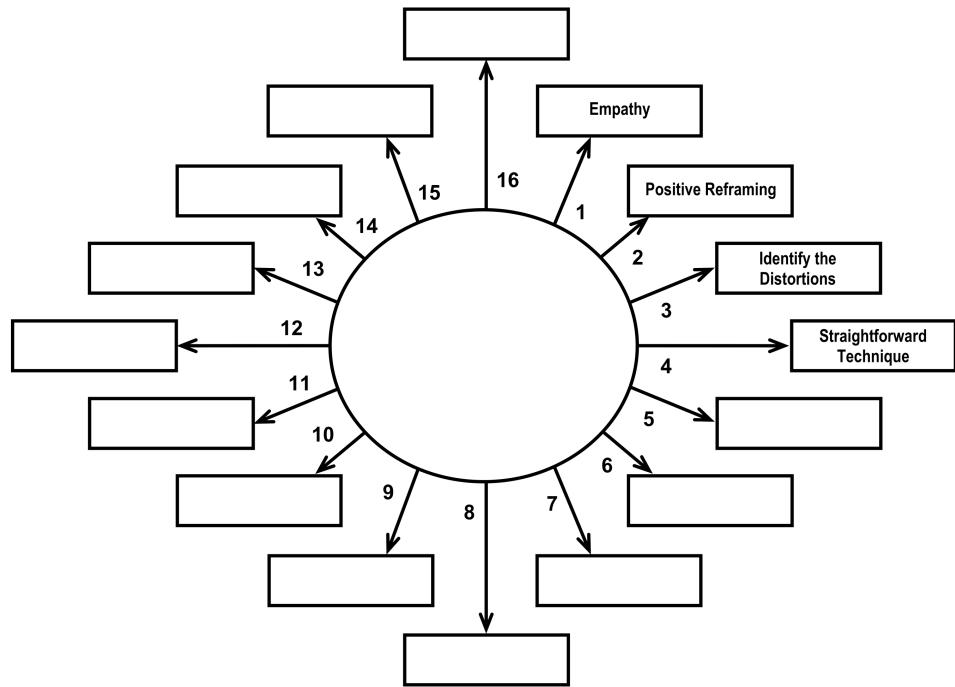
Negative Thoughts	% Now	% After	Distortions	Positive Thoughts	% Belief
6.					
7.					
8.					
9.					
10.					

Checklist of Cognitive Distortions*	
1. All-or-Nothing Thinking. You view things in absolute, black-and-white categories.	6. Magnification and Minimization. You blow things out of proportion or shrink them.
2. Overgeneralization. You view a negative event as a never-ending pattern of defeat: "This <i>always</i> happens!"	7. Emotional Reasoning. You reason from your feelings: "I <i>feel</i> like an idiot, so I must really be one."
3. Mental Filter. You dwell on the negatives and ignore the positives.	8. Should Statements. You use shoulds, shouldn'ts, musts, oughts, and have tos.
4. Discounting the Positive. You insist that your positive qualities don't count.	9. Labeling. Instead of saying, "I made a mistake," you say, "I'm a jerk" or "I'm a loser."
5. Jumping to Conclusions. You jump to conclusions not warranted by the facts. <ul style="list-style-type: none"> <li>• Mind-Reading. You assume that people are reacting negatively to you.</li> <li>• Fortune-Telling. You predict that things will turn out badly.</li> </ul>	10. Blame. You find fault instead of solving the problem. <ul style="list-style-type: none"> <li>• Self-Blame. You blame yourself for something you weren't entirely responsible for.</li> <li>• Other-Blame. You blame others and overlook ways you contributed to the problem.</li> </ul>

\* Copyright © 1984 by David D. Burns, M.D. Revised 2003.

Fig. 2. Daily Mood Journal

### **Recovery Circle\***



\* Copyright 2003 © by David D. Burns, M.D. Revised, 2018.

Fig. 3. Recovery Circle

**Positive Reframing Table\***

**Instructions.** Review the negative thoughts and feelings on your Daily Mood Log, one by one, and fill in the right-hand column. Some negative thoughts or feelings may have advantages, some may have core values, and some will have both advantages *and* core values.

Thought or Feeling	Advantages and Core Values—Ask yourself:
List each negative thought or feeling you are analyzing here. Work on them one at a time.	<p>1. What are some advantages, or benefits, of this negative thought or feeling?</p> <p>2. What does this negative thought or feeling show about me and my core values that is beautiful, positive and awesome?</p>
1. sadness and depression	<p>My depression shows my love for my son.</p> <p>My depression shows that I value connection.</p> <p>Some sadness and depression are totally appropriate.</p> <p>These feelings have motivated me to try to solve the problem of my social anxiety and to take an enormous risk in front of more than 100 people.</p> <p>These feelings show how much this matters to me.</p>
2. anxiety	My anxiety keeps me cautious and protects me and my son.
3. shame	My shame shows that I have a moral compass.
4. worthlessness	<p>My shame tells me that some changes are needed.</p> <p>My feelings of worthlessness show that</p> <ul style="list-style-type: none"> <li>• I'm honest about my shortcomings.</li> <li>• I have high standards that have motivated me to learn, to grow, and to do a great deal for my son.</li> <li>• I'm realistic about my flaws.</li> <li>• I'm accountable.</li> <li>• I'm humble and not grandiose or arrogant. Humility is a spiritual quality.</li> <li>• I'm likable.</li> <li>• I have high standards.</li> <li>• I'm aware of my shortcomings.</li> </ul>

\* Copyright © 2018 by David D. Burns, M.D. Revised, 2019.

Fig. 4. Positive Reframing

**1.3. Neuroscience Of Change**—In order to change how we think, feel and behave, we need to change certain specific networks in the brain. Those networks of nerves are the biological equivalents of thoughts. These changes can be brought about thanks to two neuroscience concepts:

- FTWT
- WTFT

Nerve cells that frequently interact with one another become connected, furthermore the more they fire together the stronger those connections get, which means that what Fires Together Wires Together (FTWT). It's not hard to notice that once we learn something it's easier to repeat the process, this comes down to the fact: nerve cells that are Wired Together tend to Fire Together (WTFT).

To understand how the brain works we can think of the SNEFF model:

- S - Structures
- N - Networks
- E - Emotions
- F - Filters
- F - Frames

Structures - different regions of our brains have different functions. The brain is divided

into structures devoted to particular tasks like the generation of emotions, the use of language, facial recognition and so on. Our focus should go to two regions: the *amygdala* and the *prefrontal cortex*. Both of these play an important role in depression and anxiety. The amygdala responds to potential threats and floods our bodies with adrenaline to prepare us for fight, flight or freeze responses. Prefrontal cortex evaluates the alarm signal from the amygdala, whether some action needs to be taken. Many studies\*,† indicate that for people who are depressed or anxious the amygdala gets activated more easily and the quality control system is less effective.

Networks - neurons (nerve cells) transmit information from one region to another. Neurons that work together are called networks. It is estimated that there are 100 billion nerve cells in the human brain and each nerve cell can connect with thousands of other nerve cells which gives 100 trillion connections between nerve cells in the brain. These neuronal networks are the physical representation of our thoughts, feelings, behaviours and most importantly they can change. Whenever we learn we are modifying those networks in our brains. This change can be brought about by the above mentioned FTWT and WTFT.

Emotions - at times they seem very powerful and overwhelming. Emotions motivate us to take actions that will increase the likelihood of survival. This emotional reactions are often automatic and they can occur so quickly that we are not even aware of what caused it but emotions are also prone to change. If our brain's quality control system identifies a better interpretation of the situation then our emotions will change and this also can occur instantly and unconsciously. We can all recall a situation when our emotions shifted when we realized that our thinking was off-base. This mechanism stands behind how CBT works, when you dispute a negative thought your interpretation of an event changes and so do your emotions. This isn't always easy, oftentimes these thoughts are feelings are very stubborn. That's partly why TEAM-CBT is so exceptional, it offers a variety of powerful methods that allow for a change to happen even if someone has suffered for years or even decades. Most importantly, these changes can often occur quickly, even in a span of an extended therapy session. Years of depression and anxiety gone in a matter of 2-3 hours.

Filters - the way we perceive the world is altered by filters, to prevent us from being overwhelmed by the amount of information our brain is receiving every second. We can't possibly pay attention to every sound, sight or smell around us. Emotions regulate what gets our attention and what gets filtered. It is not hard to notice that when we feel depressed, anxious or angry we tend to fixate on a very narrow picture of a situation and overlook the many positives that we would notice if only we felt good. Filtering easily leads to Cognitive Distortions such as all-or-nothing, overgeneralization, mental filtering, discounting the positive, mind reading, fortune telling, magnification and minimization, should statements, labeling and self-blame. These occur when the information is filtered in a negative way, even when there is no reason to do so. This can lead to a downward spiral, the more depressed we feel the more we filter information in a negative way and that makes us even more depressed. TEAM-CBT offers tools to break this cycle, identifying the distortions in each thought is one of the most effective ways.

---

\* <https://www.intechopen.com/chapters/41589>

† <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2739676/>

Frames - As mentioned above, thoughts are represented as networks of nerve cells in our brain and the connections between thoughts are the connections between networks. There is, however, not enough space in the brain to describe each event in our lives with a different set of networks, so the networks get combined, just like words to create a story. We call the combined networks - frames, they are the equivalent of the stories we tell ourselves about what's happening.

Frames and structures help us understand one of the most important concepts in TEAM-CBT - "fractal psychotherapy". This idea means to focus on a single moment when we felt upset and in that single moment all of our problems will be encapsulated and when you truly change how you were thinking and feeling at that one moment, you will understand the solution to all of your problems. That's because of frames, the negative feelings of depression, anger, inferiority, anxiety and hopelessness are the expression of frames, a set of familiar, related networks that are activated in many, if not all of your experiences. So when you change the way you are thinking and feeling at that one particular moment, you are essentially creating new networks and frames that will be activated in many other situations. Although in Daily Mood Journal our focus is narrow, the effect is extensive. The more you practice the stronger those new connections get. Let's try to integrate these concepts to TEAM-CBT.

Taking mood tests before and after each session (Testing) tells the patient and the therapist whether the micro-neurosurgery was effective and whether there is need for more.

Empathy creates a safe space for the patient to open up, share the painful thoughts and feelings and therefore activate the networks responsible for suffering, activating them is crucial in order to modify them. Empathy also helps to calm down your alarm system, to stop the flood of adrenaline, so that you can focus on the problem rather than your survival.

Agenda Setting or you can also call it Assessment of Resistance is the process of finding what's awesome and positive about your thoughts and feelings, this makes you feel more at ease with them and allows you to look at them from a different "frame", there suddenly is much less pressure to get rid of them and you can begin to feel more relaxed and in a better position to actually change them.

Methods is where the micro-neurosurgery magic happens, the patient learns to dispute the negative thought by creating more positive, realistic thoughts and by doing so creates new networks that can also be used to defeat rest of the negative thoughts.

Human brain is a fascinating organ with lots of secrets yet to be discovered, to try to understand why therapy works means trying to understand how the brain works.

*1.4. Implementation of TEAM-CBT*—The aim of this project is to take a small step in the modernization of mental health treatment by implementing the TEAM-CBT model in a web application. The system uses methods developed by Dr. David Burns and complies with the standard TEAM-CBT protocol. We also collect statistical data to evaluate whether this form is actually beneficial and works better than other forms of treatment in the standard face-to-face formula. A project of this kind gains on importance when you consider the many difficulties associated with getting some form of reliable help. There is a significant proportion of the population who do not have the access or the resources, time or money to

attend therapy.\* Moreover, not all treatments are helpful, and in many cases patients are not examined at regular intervals, so we cannot know exactly what is working and what is not. It's painful to see the extent of suffering there is around us and it's terrifying how much inadequate help is out there. The field of mental health has great potential to change and make new discoveries, after all, it is still unknown exactly why mental problems occur and why therapy is helpful or not. To ignite some hope for this approach: Bibliotherapy, which means reading a book and completing the exercises, is known to be very helpful, perhaps this digital way of trying to help will prove to be effective just as well<sup>†</sup> - "This study provided empirical evidence that a behavioral prescription for Feeling Good may be as effective as standard care, which commonly involves an antidepressant prescription".<sup>‡</sup>

## 2. System Requirements

2.1. *User Stories*—User Stories have been divided into different sections for organization purposes: General, Periodic Testing, Testing, Agenda Setting, Methods, Admin.

---

\* <https://feelinggood.com/2019/11/18/167-feeling-great-professor-mark-noble-on-team-cbt-and-the-brain/>

† <https://feelinggood.com/2017/10/04/feeling-good-bibliotherapy-does-it-really-work-or-is-it-just-hype/>

‡ <https://pubmed.ncbi.nlm.nih.gov/20803165/>

	<b>WEB APP\Testing Step User Stories</b>	
As a...	I want to...	so that...
App User	write down the specific event that caused my upset	I can focus on how I felt and what I thought in that specific event
App User	check what feelings from: Sad, blue, depressed, down, unhappy am I feeling	I know how I feel regarding the first category of feelings
App User	check what feelings from: Anxious, worried, panicky, nervous, frightened am I feeling	I know how I feel regarding the second category of feelings
App User	check what feelings from: Guilty, remorseful, bad, ashamed am I feeling	I know how I feel regarding the third category of feelings
App User	check what feelings from: Inferior, worthless, inadequate, defective, incompetent am I feeling	I know how I feel regarding the fourth category of feelings
App User	check what feelings from: Lonely, unloved, unwanted, rejected, alone, abandoned am I feeling	I know how I feel regarding the fifth category of feelings
App User	check what feelings from: Embarrassed, foolish, humiliated, self-conscious am I feeling	I know how I feel regarding the sixth category of feelings
App User	check what feelings from: Hopeless, discouraged, pessimistic, despairing am I feeling	I know how I feel regarding the seventh category of feelings
App User	check what feelings from: Frustrated, stuck, thwarted, defeated am I feeling	I know how I feel regarding the eighth category of feelings
App User	check what feelings from: Angry, mad, resentful, annoyed, irritated, upset, furious am I feeling	I know how I feel regarding the ninth category of feelings
App User	write down some other feelings I might be feeling	I know how I feel regarding the tenth category of feelings
App User	Rate in % how strongly am I feeling in each category now	I know the intensity of my feelings
App User	Rate in % how I would like to feel (after the Agenda setting step) to preserve the benefits of my negative feelings	It's easier to come to terms with my feelings
App User	Rate in % how am I feeling at the end of the session	I can see whether the session resulted in improvements
App User	write down my negative thoughts	I will be able to work on them
App User	rate how much I believe each negative thought	I will be able to compare it to the results later
App User	be able to select which cognitive distortions are present in each thought	I will be able to choose appropriate methods
App User	have an easy to look up cheat sheet with cognitive distortions	I don't have to remember all of them or check them on the internet
App User	rate how much I believe each thought after defeating it	I can see whether the selected method was effective

Table 1. User Stories for Testing

	<b>WEB APP\Periodic Tests User Stories</b>	
As a...	I want to...	so that...
App User	fill out the depression checklist anytime	I know where do I sit on a depression scale
App User	fill out the anxiety checklist anytime	I know where do I sit on an anxiety scale
App User	fill out the happiness checklist anytime	I know where do I sit on happiness scale
App User	fill out the relationships satisfaction checklist anytime	I know how satisfied I am with a particular relationship
App User	fill out the addictions checklist anytime	I know whether I struggle with some kind of addictions

Table 2. User Stories for Periodic Tests

	<b>WEB APP\Methods User Stories</b>	
As a...	I want to...	so that...
App User	fill out the recovery circle	focus on one thought and choose methods
App User	receive suggestions regarding what methods to choose	my work is more effective
App User	be able to look at quick summary of chosen technique	I will know how to use them
App User	be able to use the Straightforward Technique	I can defeat my negative thoughts
App User	be able to use the Double Standard Technique	I can defeat my negative thoughts
App User	be able to use the Examine The Evidence Technique	I can defeat my negative thoughts
App User	be able to use the Experimental Technique	I can defeat my negative thoughts
App User	be able to use the Survey Technique	I can defeat my negative thoughts
App User	be able to use the Reattribution Technique	I can defeat my negative thoughts
App User	be able to use the Socratic Method	I can defeat my negative thoughts
App User	be able to use the Thinking in Shades of Gray	I can defeat my negative thoughts
App User	be able to use the Semantic Method	I can defeat my negative thoughts
App User	be able to use the Let's Define Terms	I can defeat my negative thoughts
App User	be able to use the Be Specific	I can defeat my negative thoughts
App User	be able to use the Worst, Best, Average	I can defeat my negative thoughts
App User	be able to use the Self-Monitoring	I can defeat my negative thoughts
App User	be able to use the Worry Breaks	I can defeat my negative thoughts
App User	be able to use the Paradoxical Magnification	I can defeat my negative thoughts
App User	be able to use the Shame-Attacking Exercises	I can defeat my negative thoughts
App User	be able to use the Externalization of Voices	I can defeat my negative thoughts
App User	be able to use the Feared Fantasy	I can defeat my negative thoughts
App User	be able to use the Acceptance Paradox	I can defeat my negative thoughts
App User	be able to use the Time Projection	I can defeat my negative thoughts
App User	be able to use the Humorous Imaging	I can defeat my negative thoughts
App User	be able to use the Feared Fantasy	I can defeat my negative thoughts
App User	be able to use the Cognitive Flooding	I can defeat my negative thoughts
App User	be able to use the Individual Downward Arrow	I can defeat my negative thoughts
App User	be able to use the Interpersonal Downward Arrow	I can defeat my negative thoughts

Table 3. User Stories for Methods

	<b>WEB APP\Methods User Stories</b>	
As a...	I want to...	so that...
App User	be able to use the What-if Technique	I can defeat my negative thoughts
App User	be able to use the Hidden Emotion Technique	I can defeat my negative thoughts
App User	be able to use the Straightforward and Paradoxical Cost-Benefit Analysis	I can defeat my negative thoughts
App User	be able to use the Devil's Advocate Technique	I can defeat my negative thoughts
App User	be able to use the Stimulus Control	I can defeat my negative thoughts
App User	be able to use the Decision-Making Tool	I can defeat my negative thoughts
App User	be able to use the Daily Activity Schedule	I can defeat my negative thoughts
App User	be able to use the Pleasure-Predicting Sheet	I can defeat my negative thoughts
App User	be able to use the Anti-Procrastination Sheet	I can defeat my negative thoughts
App User	be able to use the Gradual Exposure and Flooding	I can defeat my negative thoughts
App User	be able to use the Response Prevention	I can defeat my negative thoughts
App User	be able to use the Distraction	I can defeat my negative thoughts
App User	be able to use the Image Substitution	I can defeat my negative thoughts
App User	be able to use the Memory Rescripting	I can defeat my negative thoughts
App User	be able to use the Smile and Hello Practice	I can defeat my negative thoughts
App User	be able to use the Talk Show Host	I can defeat my negative thoughts
App User	be able to use the Self-Disclosure	I can defeat my negative thoughts
App User	be able to use the Rejection Practice	I can defeat my negative thoughts
App User	be able to use the Blame/Relationship Cost-Benefit Analysis	I can defeat my negative thoughts
App User	be able to use the Relationship Journal	I can defeat my negative thoughts
App User	be able to use the Five Secrets of Effective Communication	I can defeat my negative thoughts
App User	be able to use the One-Minute Drill	I can defeat my negative thoughts

Table 4. User Stories for Methods

	<b>WEB APP\Agenda Setting User Stories</b>	
As a...	I want to...	so that...
App User	write down the positives for each of the category of feelings and for each thought	I can melt away my resistance to change
App User	be able to receive suggestions regarding positives	I can receive help when I'm stuck

Table 5. User Stories for Agenda Setting

	<b>WEB APP\General User Stories</b>	
As a...	I want to...	so that...
App User	be able to look through my previous sessions	I can see how I tackled a particular problem in the past
App User	continue a previous session	I don't have to complete a whole session at once
App User	filter the previous sessions based on time	I can faster find the session of interest
App User	be able to set up an account	I can save my results in the database
App User	have my results saved in the database	I don't lose my data
App User	manage my account	I can change my information, for example: password
App User	log out	I can keep my information safe from others using the same device
App User	have access to a pleasing UI	I am more eager to use the application
App User	access the web application on multiple devices and browsers	I have flexibility in regards to what I use to access the website
App User	be able to log in	I can have access to my sessions and tests

Table 6. User Stories - General

	<b>Admin</b>	
As a(n)...	I want to...	so that...
Admin	have access to see the statistics of total user base	I can provide a better service to the users
Admin	see the statistics of a particular user	I can see whether the user is improving and make some recommendations
Admin	restrict the statistics data by time	I can have a look at the data from a particular period
Admin	be able to block the user accounts if it's needed	I can defend the system from malicious users (e.g.: bots filling the tests)
Admin	reject some tests	I can prevent the unreliable data
Admin	filter the tests based on time and userID/email	I can easier find the tests I need to check
Admin	access the web application on multiple devices and browsers	I have flexibility in regards to what I use to access the website

Table 7. User Stories for Admin

**2.2. Critical Requirements**—We present the critical requirements in the form of reverse side of user stories, that also include non functional requirements.

	<b>User Story</b>	
As a...	I want to...	so that...
App User	fill out the recovery circle	focus on one thought and choose methods
	<b>Acceptance Criteria</b>	
	<ul style="list-style-type: none"> <li>-Can I choose 13 different methods?</li> <li>-Can I look up information about each method?</li> <li>-Can I change my choice?</li> <li>-Can I search for the method I want to choose by using text search?</li> </ul>	
	<b>Non Functional Requirements</b>	
	<ul style="list-style-type: none"> <li>-Is the method added in less than 1 second?</li> </ul>	

Table 8. Reverse Side for "fill out the recovery circle"

	<b>User Story</b>	
As a...	I want to...	so that...
App User	be able to look at quick summary of chosen technique	I will know how to use them
	<b>Acceptance Criteria</b>	
	<ul style="list-style-type: none"> <li>-Can I choose any of the implemented methods and see a summary?</li> </ul>	
	<b>Non Functional Requirements</b>	
	<ul style="list-style-type: none"> <li>-Can I look up the summaries at any point of going through TEAM?</li> <li>-Is the summary displayed in an aesthetic way?</li> </ul>	

Table 9. Reverse Side for "be able to look at quick summary of chosen technique"

	<b>User Story</b>	
As a...	I want to...	so that...
App User	receive suggestions regarding what methods to choose	my work is more effective
	<b>Acceptance Criteria</b>	
	<ul style="list-style-type: none"> <li>-Can I receive suggestion while choosing methods for recovery circle?</li> <li>-Are the suggestions based on distortions I have found in my thought?</li> <li>-Are the suggestions based on whether a particular method was useful to me in the past?</li> </ul>	
	<b>Non Functional Requirements</b>	
	<ul style="list-style-type: none"> <li>-Is the suggestion displayed in under 500ms?</li> <li>-Can I choose to auto-fill whole recovery circle?</li> </ul>	

Table 10. Reverse Side for "receive suggestions regarding what methods to choose"

	<b>User Story</b>	
As a...	I want to...	so that...
App User	be able to use the ... Technique	I can defeat my negative thoughts
	<b>Acceptance Criteria</b>	
	<ul style="list-style-type: none"> <li>-Can I add the ... Technique to the recovery circle?</li> <li>-Can I use the ... Technique?</li> </ul>	
	<b>Non Functional Requirements</b>	
	<ul style="list-style-type: none"> <li>-Is the ... Technique provided to me in an interactive way?</li> <li>-Is the implementation lightweight, are the animations smooth, if any?</li> </ul>	

Table 11. Reverse Side for "be able to use the ... Technique"

	<b>User Story</b>	
As a...	I want to...	so that...
App User	write down the specific event that caused my upset	I can focus on how I felt and what I thought in that specific event
	<b>Acceptance Criteria</b>	
	-Can I fill a text box with my event?	
	<b>Non Functional Requirements</b>	
	-Can I receive information how should I describe the event?	

Table 12. Reverse Side for "write down the specific event that caused my upset"

	<b>User Story</b>	
As a...	I want to...	so that...
App User	check what feelings from: ... am I feeling	I know how I feel regarding the ... category of feelings
	<b>Acceptance Criteria</b>	
	-Can I select the feelings I'm experiencing? -Can I write down the % of belief in each category?	
	<b>Non Functional Requirements</b>	
	-Can I change my selection? -Is the website pleasing to the eye?	

Table 13. Reverse Side for "check what feelings from: ... am I feeling"

	<b>User Story</b>	
As a...	I want to...	so that...
App User	write down my negative thoughts	I will be able to work on them
	<b>Acceptance Criteria</b>	
	-Can I edit each of written thoughts during the whole process of writing them down? -Can I add any number of thoughts?	
	<b>Non Functional Requirements</b>	
	- Does the page function smoothly after adding a big number of thoughts(for example 20)?	

Table 14. Reverse Side for "write down my negative thought"

	<b>User Story</b>	
As a...	I want to...	so that...
App User	have an easy way to look up a cheat sheet with cognitive distortions	I don't have to remember all of them or check them on the internet
	<b>Acceptance Criteria</b>	
	-Can I have access to them at any time during the session?	
	<b>Non Functional Requirements</b>	
	-Can the cheat sheet be translated to a different language? -Can I download the cheat sheet?	

Table 15. Reverse Side for "have an easy way to look up a cheat sheet with cognitive distortions"

	<b>User Story</b>	
As a...	I want to...	so that...
App User	fill out the depression/anxiety/relationships/happiness/addiction checklist anytime	I know where do I sit on a depression/anxiety/relationships/happiness/addiction scale
	<b>Acceptance Criteria</b>	
	<ul style="list-style-type: none"> <li>-Is the test provided to me in an aesthetic, interactive way?</li> <li>-Can I see the result right after submitting the checklist?</li> <li>-Can I have a reminder to fill in the checklist on a regular basis to track progress?</li> </ul>	
	<b>Non Functional Requirements</b>	
	<ul style="list-style-type: none"> <li>-Can I receive a monthly report to see how did my month of check lists go?</li> <li>-Can I print the results after receiving it?</li> </ul>	

Table 16. Reverse Side for "fill out the depression/anxiety/relationships/happiness/addictions checklist"

	<b>User Story</b>	
As a...	I want to...	so that...
App User	manage my account	I can change my information
	<b>Acceptance Criteria</b>	
	<ul style="list-style-type: none"> <li>-Can I edit all the fields, besides login, that I have provided during the registration?</li> <li>-Can I save my newly edited information?</li> </ul>	
	<b>Non Functional Requirements</b>	
	-Is the new information saved in the database in under 1 second?	

Table 17. Reverse Side for "manage my account"

	<b>User Story</b>	
As a...	I want to...	so that...
App User	be able to set up an account	I can save my results in the database
	<b>Acceptance Criteria</b>	
	<ul style="list-style-type: none"> <li>-Can I fill in the necessary information - email and password?</li> <li>-Can I provide additional information such as age or gender?</li> </ul>	
	<b>Non Functional Requirements</b>	
	<ul style="list-style-type: none"> <li>-Is the process of registration secure?</li> <li>-Is it checked whether the user already exists in the database?</li> <li>-Is the new user added to the database in under 1 second?</li> </ul>	

Table 18. Reverse Side for "be able to set up an account"

	<b>User Story</b>	
As a...	I want to...	so that...
App User	be able to log in	I can have access to my sessions and tests
	<b>Acceptance Criteria</b>	
	-Can I enter my credentials? -Can I receive error information in case the credentials are incorrect?	
	<b>Non Functional Requirements</b>	
	-Is the login secured, for example by using OAuth2?	

Table 19. Reverse Side for "be able to login"

	<b>User Story</b>	
As a...	I want to...	so that...
Admin	see the statistics of taking a particular test	I can see with what users struggle the most
	<b>Acceptance Criteria</b>	
	-Can I generate a chart for each test? -Can I limit the data to a particular user? -Can I select a particular time period?	
	<b>Non Functional Requirements</b>	
	-Can I download the selected statistics in a pdf?	

Table 20. Reverse Side for "see the statistics of taking a particular test"

	<b>User Story</b>	
As a...	I want to...	so that...
Admin	have access to see the total data in a chart	I can provide a better service to the users
	<b>Acceptance Criteria</b>	
	-Can I see the combined data of mood tests of all users? -Can I see a chart? -Can I select a particular time period?	
	<b>Non Functional Requirements</b>	
	-Is page pleasing and nicely formatted?	

Table 21. Reverse Side for "have access to see the total data in a chart"

	<b>User Story</b>	
As a...	I want to...	so that...
Admin	reject some tests	I can reject the unreliable data
	<b>Acceptance Criteria</b>	
	<ul style="list-style-type: none"> <li>-Can I have access to a database with all the tests?</li> <li>-Can I delete selected entries?</li> <li>-Can I select a particular time period?</li> </ul>	
	<b>Non Functional Requirements</b>	
	<ul style="list-style-type: none"> <li>-Can I create a backup of removed data?</li> <li>-Is the operation done swiftly, in under 1 second?</li> </ul>	

Table 22. Reverse Side for "reject some tests"

2.3. *Use Cases*—Below we present Use Case Diagrams that have been divided into sections:

- General - Here user can start a new session, take a mood test (an evaluation), continue previous session, manage account, login, logout or register.
- Testing - this corresponds to the T - Testing step, user can write down the upsetting event, mark the feelings and %, write down upsetting thoughts and find distortions in them.
- Paradoxical Agenda Setting - User is able to write the benefits of previously selected thoughts and feelings and receive suggestions.
- Methods - gives the possibility to the user to choose the most upsetting thought for the recovery circle, choose the methods and go through them.
- Admin - Admin is able to check test (Evaluation) result and optionally reject the test, in case the system is flooded with fake tests. Admin is also able to manage user accounts and see the statistical data, whether the users are improving and whether the system is helpful.

More detailed explanation of each action is presented later on in the document.

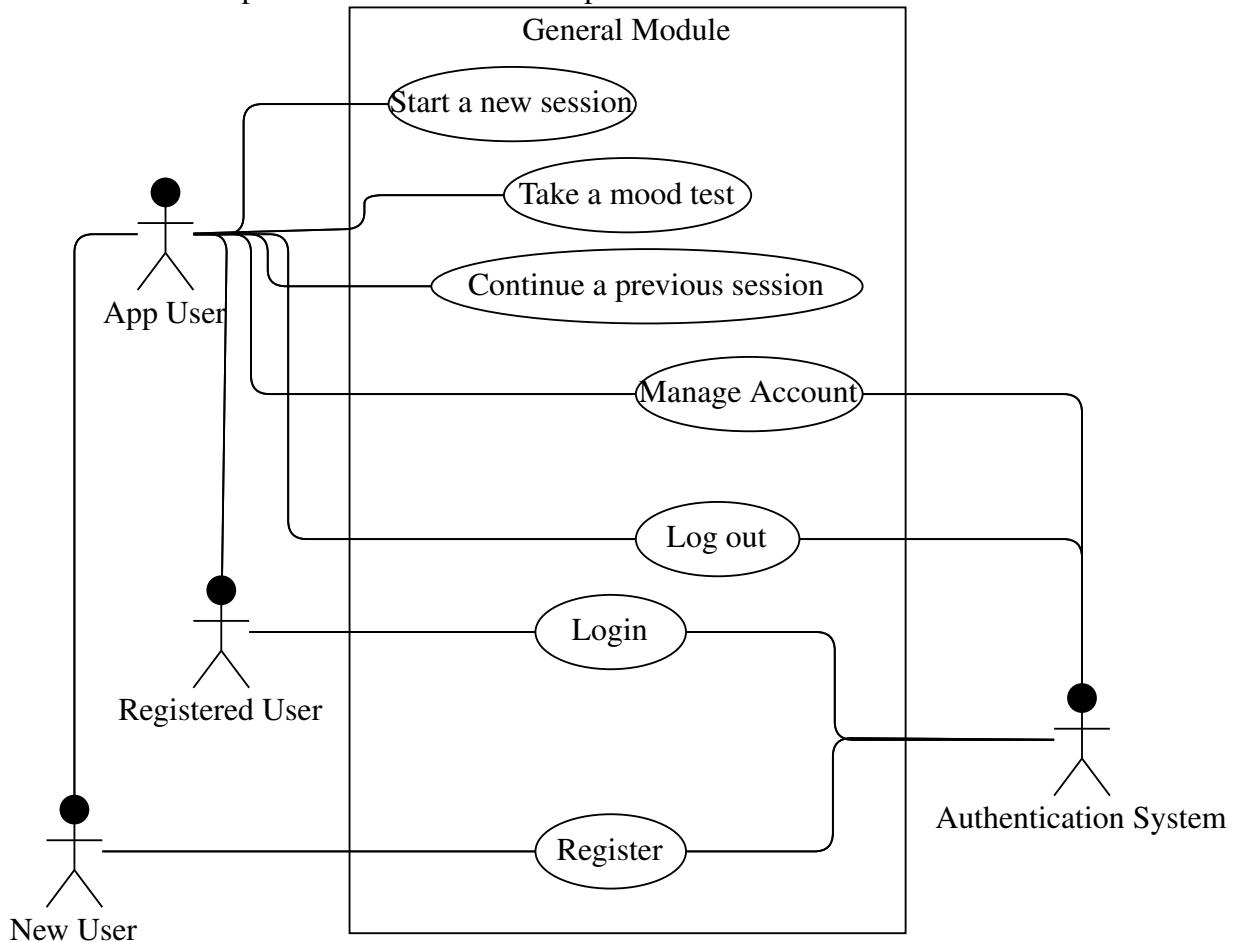


Fig. 5. Use Case Diagram for General module

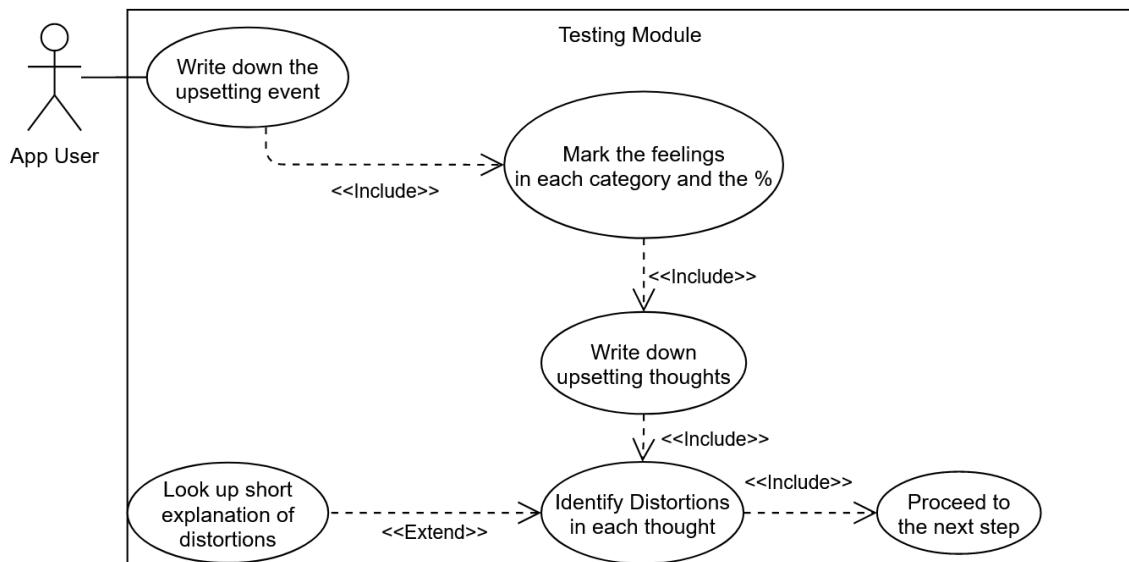


Fig. 6. Use Case Diagram for Testing module

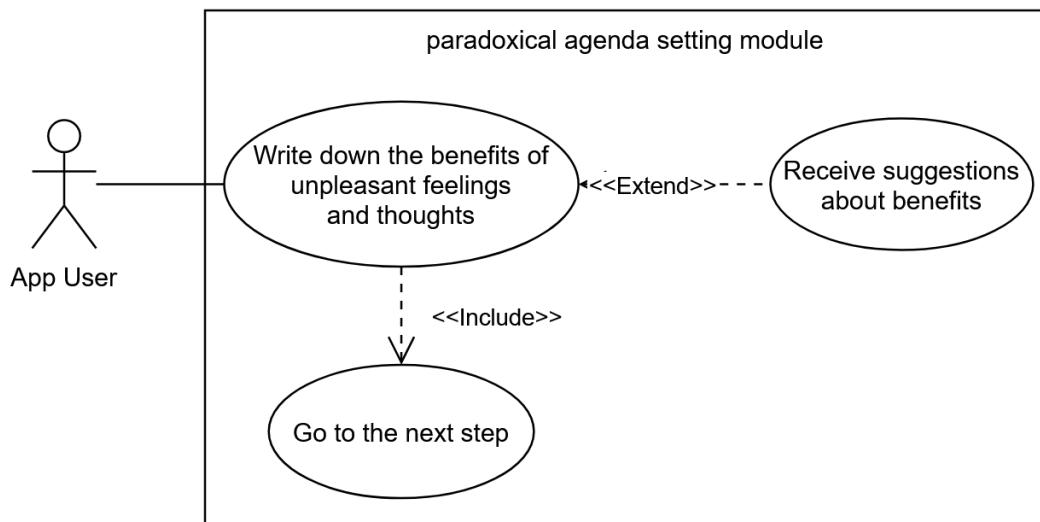


Fig. 7. Use Case Diagram for Agenda module

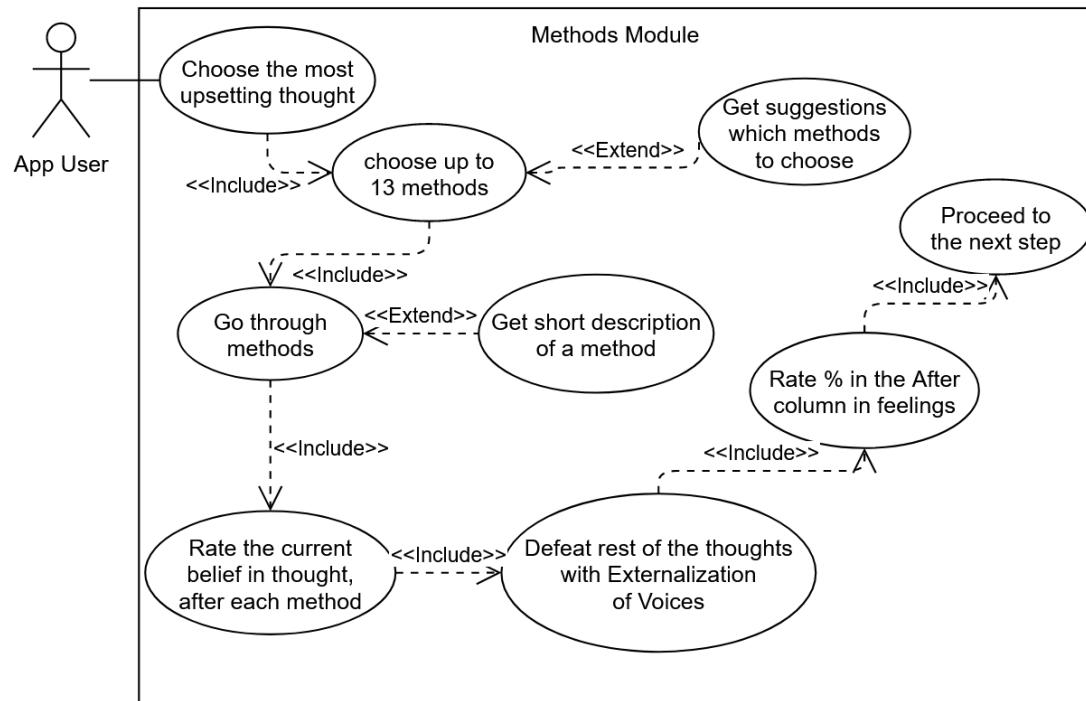


Fig. 8. Use Case Diagram for Methods module

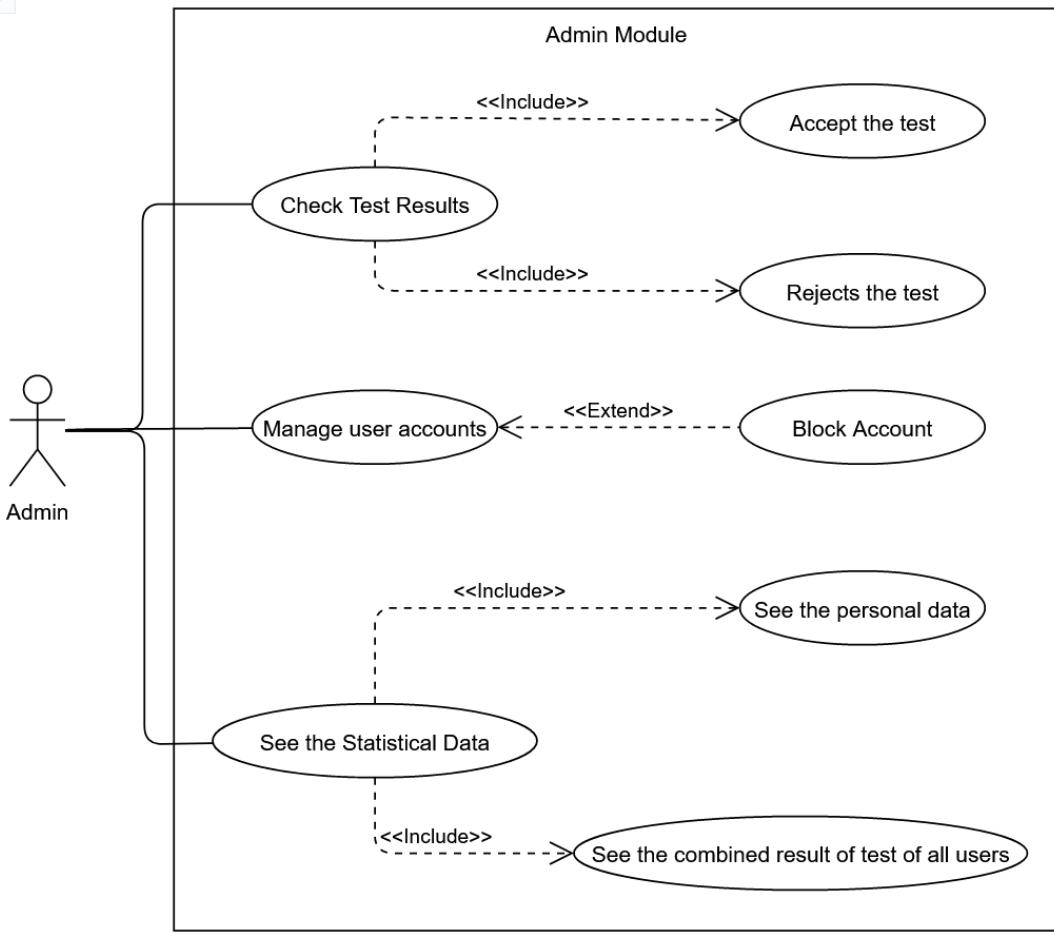


Fig. 9. Use Case Diagram for Admin module

### 3. Architecture Design

We propose a system consisting of a server, database and web application with two different interfaces for users and for an admin. The communication between the web application and the server will be realised with an API presented in further sections.

Although a mobile app may seem to be a more natural direction for this kind of idea, we chose a web app for the following reasons:

- an app utilizing TEAM-CBT is already in development
- a website should reach more users than an app
- it is easier to update the site often compared to an app

Web Application provides a friendly interface to the user that will allow to go through the TEAM steps with ease and also to take mood tests whenever the user feels it is reasonable to do so, it would be best to take tests at weekly intervals, however it is not required. The web application also provides an interface for the admin, that will be able to manage user's accounts, for example an admin might ban a user that seems to be a bot and is filling hundreds of tests. The admin can also see the statistical data regarding the user's progress in terms of their mood test (Anxiety, Depression, Addiction, Happiness, Relationship) scores.

Server is responsible for performing operations needed by the web app to function properly, such as performing calculations needed for statistical data or providing suggestions as to which methods will be effective for a user based on cognitive distortions found in a particular thought and perhaps based on their previous sessions.

The database holds data about statistics, registered users, user sessions and information needed to conduct a session such as:

- information about methods, description of them which are stored as simple strings of words
- information about cognitive distortions
- available evaluations and question sets

## 4. System Objects

In the whole system we can distinguish some essential objects representing elements necessary to conduct a session or to go through an evaluation.

- Session - Corresponds to the real-life process of going through therapy session that complies to the TEAM protocol. The whole idea of a session is represented in the system by two classes: *Session* and *SessionService*. The attributes of *Session* class represent data needed to implement such functionality. *Session* holds: *MethodModule*, *TestModule* and *AgendaModule*. These modules correspond to the TEAM steps (a reminder that E - empathy step is omitted) and contain attributes that are crucial for each step to be conducted properly.
- Thoughts - A big part of going through session is writing down thoughts, whether negative or positive, and trying to think of them from different perspectives with the help of different methods. In the system the thoughts are represented by a class *Thoughts* holding arrays of *PositiveThought* and *NegativeThought*. *PositiveThought* class holds content of the thought in a form of string, name of the method used to generate that thought, reduction in belief in negative thought that this positive thought caused and the belief in the thought itself. *Negative Thought* class holds a list of positive thoughts that were generated to counter this negative thought, initial % of belief in the thought, a string representing the content of negative thought, selected distortions of type enum *CognitiveDistortions* and an array holding the % of belief after using each method.
- Evaluation - In order to know if any of these techniques work, we need to somehow test users' mood. This can be done by mood tests, these tests are represented by *Evaluation*(name Test will most likely be reserved for other purposes). All the essential information is contained in this class: date of when the evaluation was taken, status of the evaluation, a string indicating if the evaluation is *Not Started*, *In Progress*, *Finished*, *Rejected*. We offer several test categories: Depression, Anxiety, Relationship Satisfaction, Addictions, Happiness. The user doesn't have to be taking all of them at once or at all. However it is highly recommended to be taking the first two tests at regular intervals, every week or so.
- Method - methods are necessary to defeat negative thoughts, they are simple cues, ideas that prompt the user in the right direction to come up with a powerful positive

thought that will put the lie to the negative thought. There is a wide range of methods (not all of them are represented on the class diagram) so user need not to worry about whether a certain method will work or not, there is always another one. Methods are represented by Method class that contains: name of the method, reduction in belief in the negative thought that this method caused and description of the method containing all the necessary information needed for the user to use it.

- User - User, which in this context could also be called a patient, is represented in the system as a *User* class. It holds user's unique ID, that will be used for many functionalities of the system, user's sessions as a list, similarly for evaluations and a banned attribute, that indicates whether the user has been banned by the admin or not.
- Statistics - Represented by *Statistics* class. It contains an array for each of the types of evaluations holding the results, result is a simple integer and an ID, each entry of the array represents a different point in time when the evaluation has been taken, these time points are saved in date attribute, this class also holds userID that is associated with the statistics, if the statistics are of general kind (concerning whole user base) the userID is set to -1.

## 5. System Structure

The system can also be divided into **backend**, that is responsible for necessary computations that provide the functionality to the user and **frontend** which is the part of the system that the user directly interacts with.

We wanted to ensure the software's high quality, fast development and maintainability, therefore we followed the principles of object-oriented design and so each of the presented classes plays a particular role.

- Classes such as *Thoughts*, *User* or *Evaluation* only contain the necessary attributes without methods. They can be thought of as **Data Transfer Object** classes, they reduce the cost of data transfer by aggregating the data that would otherwise be transferred by several calls. These classes also represent the tables contained in the database.
- On the diagram there are also **Business Domain** classes holding attributes and methods needed to implement to implement a particular functionality like *MethodsStep* class.
- Server and database need to communicate with each other, *Service* classes are made just for that. They hold the functions needed to perform actions on particular area of the system, for example *Session* and *SessionService*.
- To allow for communication between frontend and server *APIController* has been defined. This class defines HTTP requests - GET, POST, PUT, DELETE.
- Cognitive Distortions are represented by an *enum*, with twelve (two of the distortions have been split, that's why it's twelve instead of ten) different values, each value corresponding to a different distortion.
- Representation of the database is also on the diagram - *DB\_Main*. This class contains

There are also types of classes that have been omitted from the diagram such as *Data Access Object* or *Repositories*

Two relevant sections of the diagram can be distinguished. The frontend with different Panels. The backend with a APIController and a connection to the database that allows the data about users, evaluations and session be saved in backend.

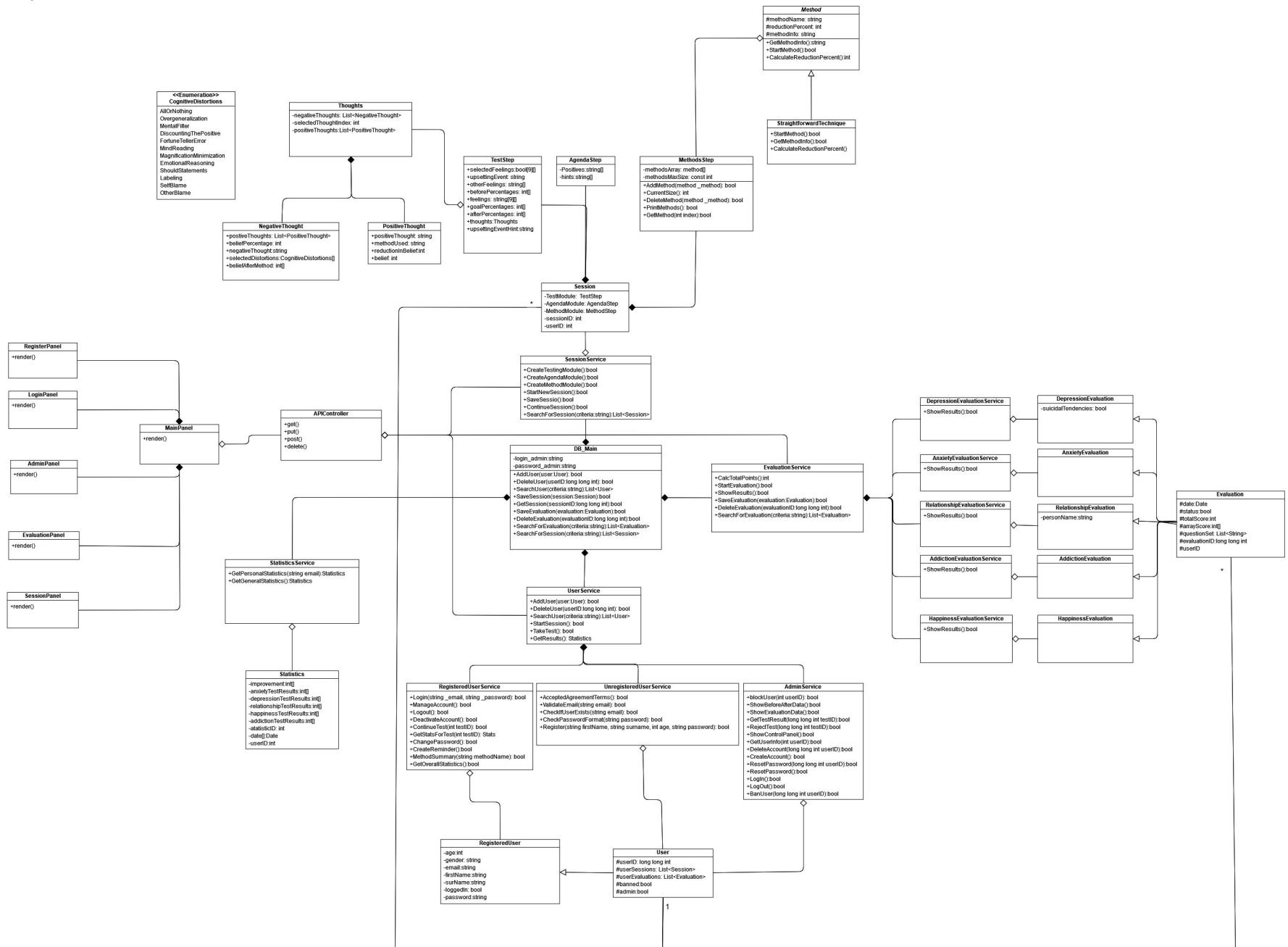


Fig. 10. Class Diagram

## 6. System States

System objects can be in different special states and in each of these states a unique behaviour will be applied by the system. Below the State Diagrams are presented for:.

For User, following states will be used:

- Not\_Logged\_in - in this state the user sees a screen three buttons: login, register and start a guest session. Each of the buttons leads to a different page, login page, registration page and session page respectively.
- Registering - user is in the process of registering. Several fields need to be filled in like unique email, that is not already registered, password and optionally age and gender. When the registration process is successful, the data is saved in the database.
- Logging\_in - user sees a screen with two empty fields - email and password and a button that sends the login request to the server.
- Recovering\_password - user is on the recover password page, an email field needs to be entered and then if such email is registered (such account exists in the database) a link with recovery website is sent on the provided email, from where user can enter a new password and PUT request is sent to the server to update account's data.
- Failed\_login - User failed to enter correct credentials more than 3 times, the possibility to log in is blocked for 2 minutes in order to stop any brute-force attacks. Information about that is shown on the page. In this state it is possible to choose to recover password.
- Logged\_in - User successfully logged in and now sees a screen with buttons allowing for starting a new session, that will be assigned to that account, starting an evaluation, continuing previous session or managing account.

Session can be in the below states:

- Not\_Start - Session is empty, the user hadn't had a chance to interact with it yet.
- Started\_Testing - Session has been started and is now in T - *Testing* step. Testing module is being shown on the screen, that consists of the same fields that are present in the, previously mentioned, Daily Mood Journal. Data is being collected and sent to the database. User enters the upsetting event, selects and rates the feelings. But also writes down the negative thoughts, rates the % belief in each thought and finds the Cognitive Distortions.
- Started\_Agenda - *Agenda* setting module is being shown on the screen. User writes down the positives of selected negative feelings per category (there are 10 categories as in the Daily Mood Journal) and the positives of negative thoughts. User has the possibility to receive suggestions what positives could be found in each category of feelings, these positives are rather generic and can be a previously prepared list, every user can see the same list. At the end of this process user writes down the desired percentage of negative feelings (the Goal% column).
- Started\_Methods - *Methods* step is started and the appropriate module is displayed. User chooses one thought to work with, preferably the most upsetting one, and chooses 13 methods. User has the possibility to receive suggestions, as to which methods to choose based on the Cognitive Distortions found in the thought. These suggestions are also rather generic, a table with appropriate suggestions can be found in Appendix A.

After the thought has been defeated (is no longer that upsetting, ideally not upsetting at all) user can defeat the rest of the thoughts, this time without Recovery Circle, just by using the Externalization of Voices, which means writing down more positive thoughts according to some simple guidelines.

- Started\_Testing\_Finish - User concludes the session by marking the change in all category of feelings (After%). At this point it is evident whether the session was helpful or not.
- Finished - The screen with a summary is shown, user can see the upsetting event, change in each category of feelings and the negative thoughts with a corresponding positive thought.

Server's main state is **Waiting for requests** from there it can go to states such as: **Getting previous sessions**, **Getting user's details** or **Getting Evaluations**, in these states, server connects to the database and tries to fetch desired data, returns an error message to the frontend in case of failure and data in case of success. Similar states are created for updating and creating new entries in the database (not depicted on the diagram). Evaluation can exist in the following states:

- Not\_Started - Evaluation has not been started yet.
- In\_Progress - The question set is being shown on the screen and user can answer each question with a number (1-4), the questions sets are presented in Appendix B, collected data is being saved to the database.
- Finished - Summary screen is shown to the user, where the information about the total score and the meaning behind it can be seen. These information can also be found in Appendix B.

Frontend's state diagram starts with Start Page, where a user can login or register. In case of deciding to set up a new account a register page is displayed, where user can enter the required data, if the validation of register request was successful a login page is displayed where the user needs to enter correct credentials. After logging in, it is checked whether the user is an admin, depending on that two different interfaces are displayed:

1. Admin Main Page - from here admin can:

**Check test results** - a page with a list of tests is displayed, after being fetched from the backend, the test can be filtered based on time period and admin can select a number of them to reject those that are deemed invalid.

**Manage Accounts** - Users are fetched from the database and a list of them is displayed. Admin can press on any user to display details. It is also possible to search for a particular user based on userID/email. In the user's detail page, admin has a possibility to ban that particular user.

**See Statistics** - A page with two options is displayed - either to see personal or general statistics. In general statistics page admin is presented with a chart (that can be modified by selecting a time period) that shows the results of mood tests of the whole user base (excluding banned users). If the admin decides to see personal statistics and provides userID/email a page with similar chart is displayed that shows data for a specific user.

**Logout** - Admin gets logged out and is presented with Start Page.

2. User Main Page - user has the possibility to:

**Start a New session** - session data is fetched from backend and a page where user can go through the TEAM steps is displayed.

**Continue previous session** - a list of sessions associated with that particular user is fetched from the database (with pagination), user can restrict the time period and click on one of the sessions. When the session is clicked a Continue Session page is displayed, where user can go through session starting from where the session has been saved. If the session is in *Finished* state then the Testing step is displayed and user can freely move between steps and edit data.

**Manage Account** - A page with user's details, after being fetched from database, is displayed. User can freely edit data besides username (email).

**Take A Mood Test** - User enters Take A Mood Test page where a list of mood test types (depression, anxiety, relationships, addictions, happiness) id displayed. After clicking on one of the test types a Mood Test page is shown, where user can go through the selected test and immidietely see the results.

**Logout** - User gets logged out and is presented with Start Page.

In nearly all of the above frontend pages user/admin has the possibility to go back to the previous page by pressing a button, it is depicted on the below diagram.

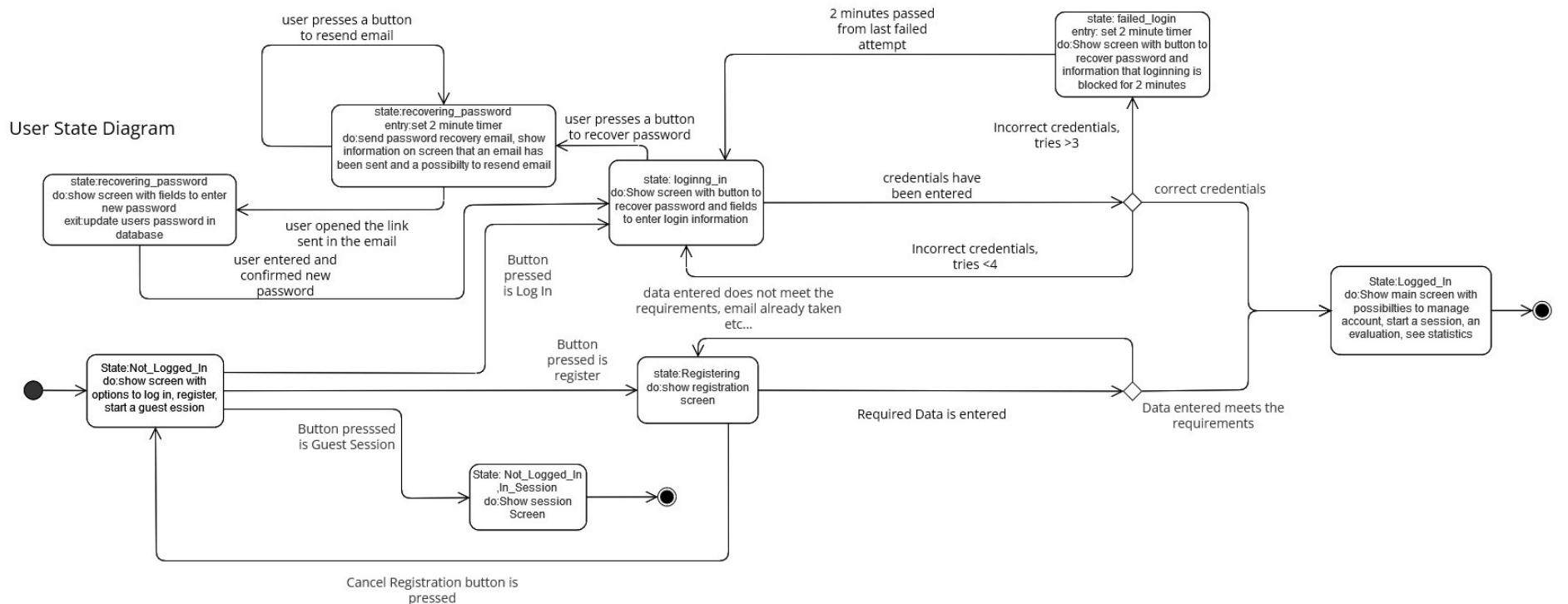


Fig. 11. State Diagram for User

#### Session State Diagram

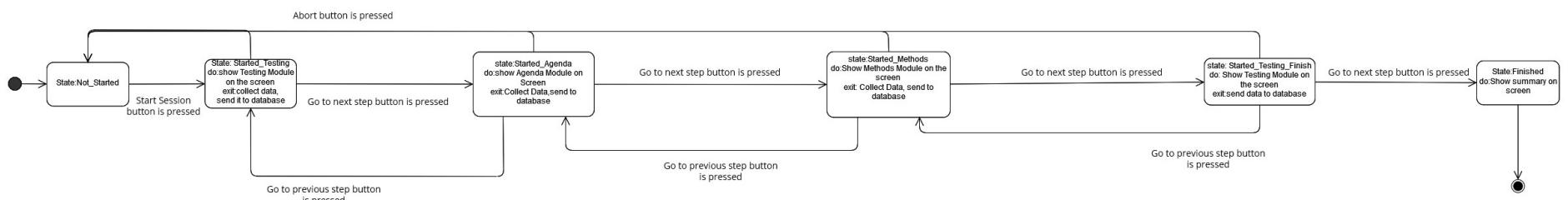


Fig. 12. State Diagram for Session

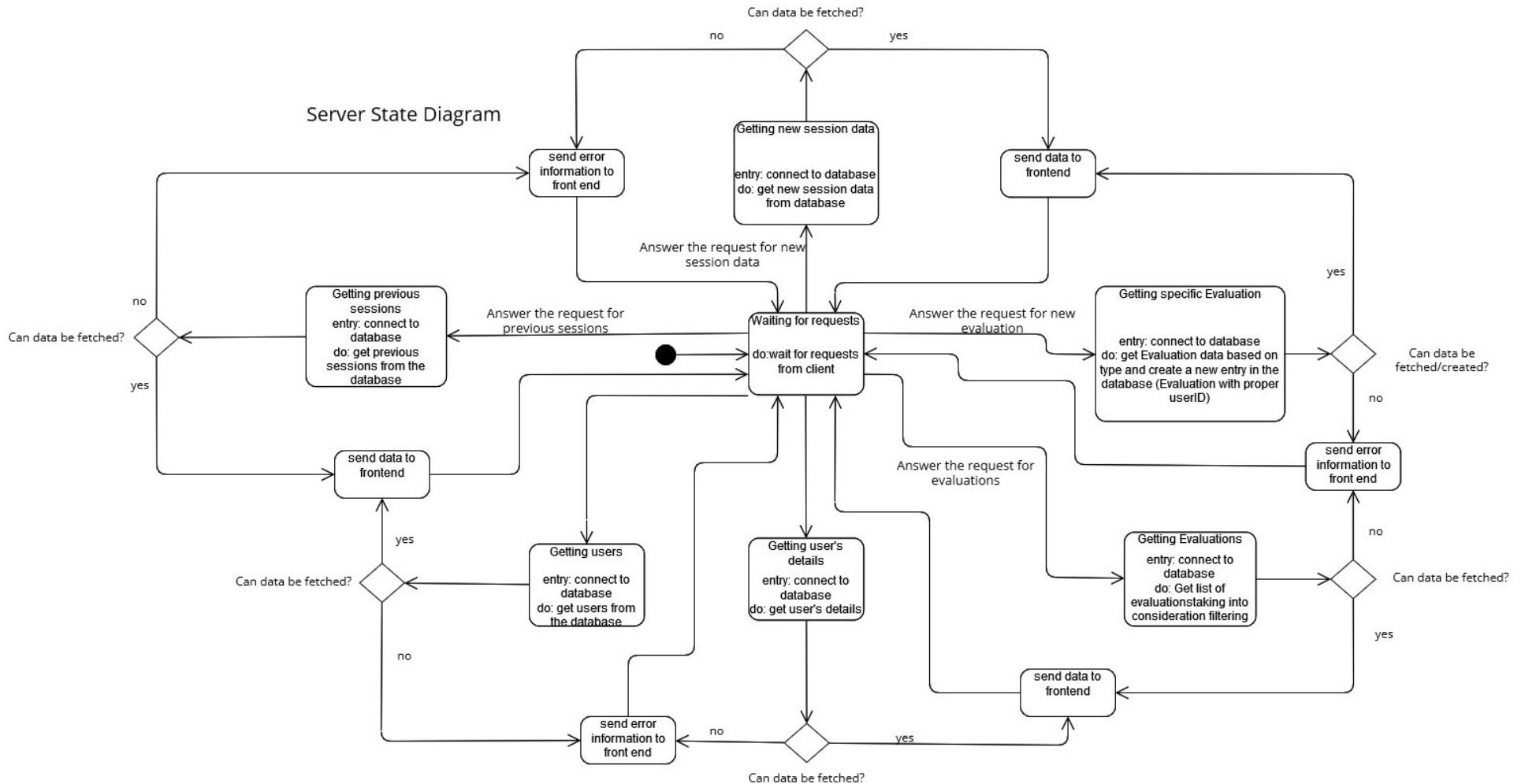


Fig. 13. State Diagram for Server

## Evaluation State Diagram

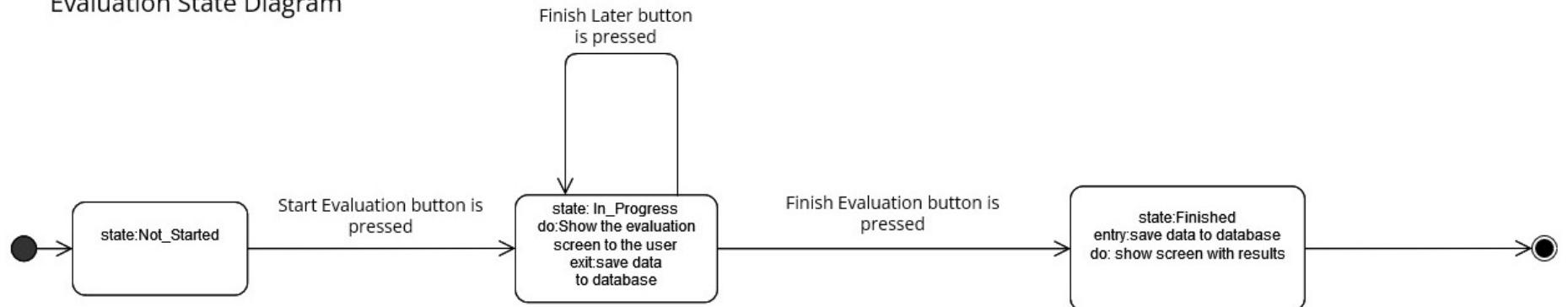


Fig. 14. State Diagram for Evaluation

## Frontend

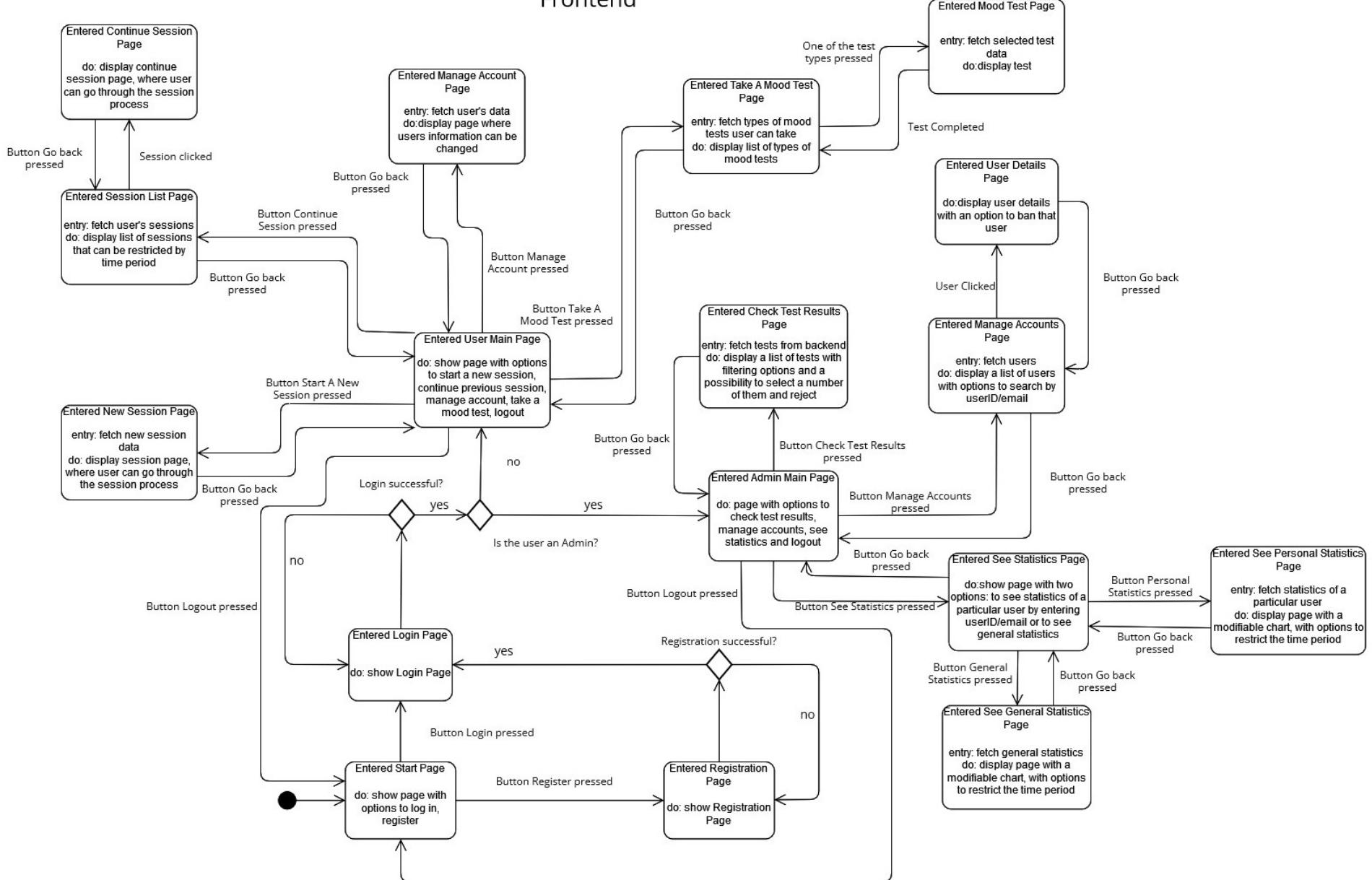


Fig. 15. State Diagram for frontend

## 7. System Activity

Actions that can be performed by users and admins have been represented as activity diagrams, they are prepared for essential parts of the system.

**Login** - In order to log in user first needs to press the login button (other choices are: register and guest session), then the login page is displayed where the user can enter credentials, email and password, a *POST* request is sent to the server containing email and password, data is verified with that contained in the database, if the user fails to enter proper data 3 times a 2 minute blockade is enabled. After successful login, user is presented with a home page that brings the possibility to start a new session, continue a preview session, take a mood test and to manage account.

**New Session** - After pressing the new session button data needed to start a new session is being retrieved from the *backend* and the user goes through the TEAM steps, meanwhile data is being saved to the database, at the end of the process, user can have a look at the summary screen, showing the work done and it's effects.

**See Statistics** - Admin presses see statistics button and is given a choice between being able to see statistics of a specific account or combined, overall statistics. This enables administrator to examine effectiveness of the app.

**Register** - In order to have access to previous sessions and statistics user needs to have an account, and so user is offered with the possibility to register. The process starts with user pressing the register button, then the user is presented with a page where an email and a password, together with some optional fields like age or gender, should be entered. If an account with the same email already exists the registration fails. The fields are also validated at the *frontend* in terms of correct format. If the registration is successful the user is presented with a login page.

**Start Evaluation** - An important part of the system are tests that can be taken at anytime, preferably at regular intervals. After clicking evaluation button, user can choose between Depression, Anxiety, Addictions, Relationships and Happiness tests. Questions are being retrieved from the *backend* and user answers each question with a number in the range 0-4. At the end total score is displayed with an explanation of what such score means.

**Continue Previous Session** - When the continue previous session button is pressed, a list of sessions with a state different than *Not\_started* is retrieved from *backend* (with pagination) and displayed on *frontend*. User can choose any session and then the session is displayed from the start (T step), the data can be edited and so the summary will also change.

**Ban User** - Admin has the possibility to ban a user that, for example might be a bot. To do so he presses manage user accounts button, where a list of users is displayed, that also can be filtered based on *userID* or user's *email* and then any user can be banned.

**Manage Account** - Each user has the possibility to edit some account information. To do so the Manage Account button needs to be pressed, data is retrieved from the database and manage account page is shown on the screen, user can then edit desired fields and confirm them by pressing save button, then *UPDATE* request is sent to the server and new data is saved in the database. Finally manage account page is shown with newly updated data.

**Show Test Results** - When admin presses show test results button, a list of tests is displayed, the list can be filtered based on time period and *userID/email*. When a test is

selected and reject button is pressed, the *status* field of that test is changed to *rejected* and it is no longer taken into consideration during computation of statistics.

### Login Process

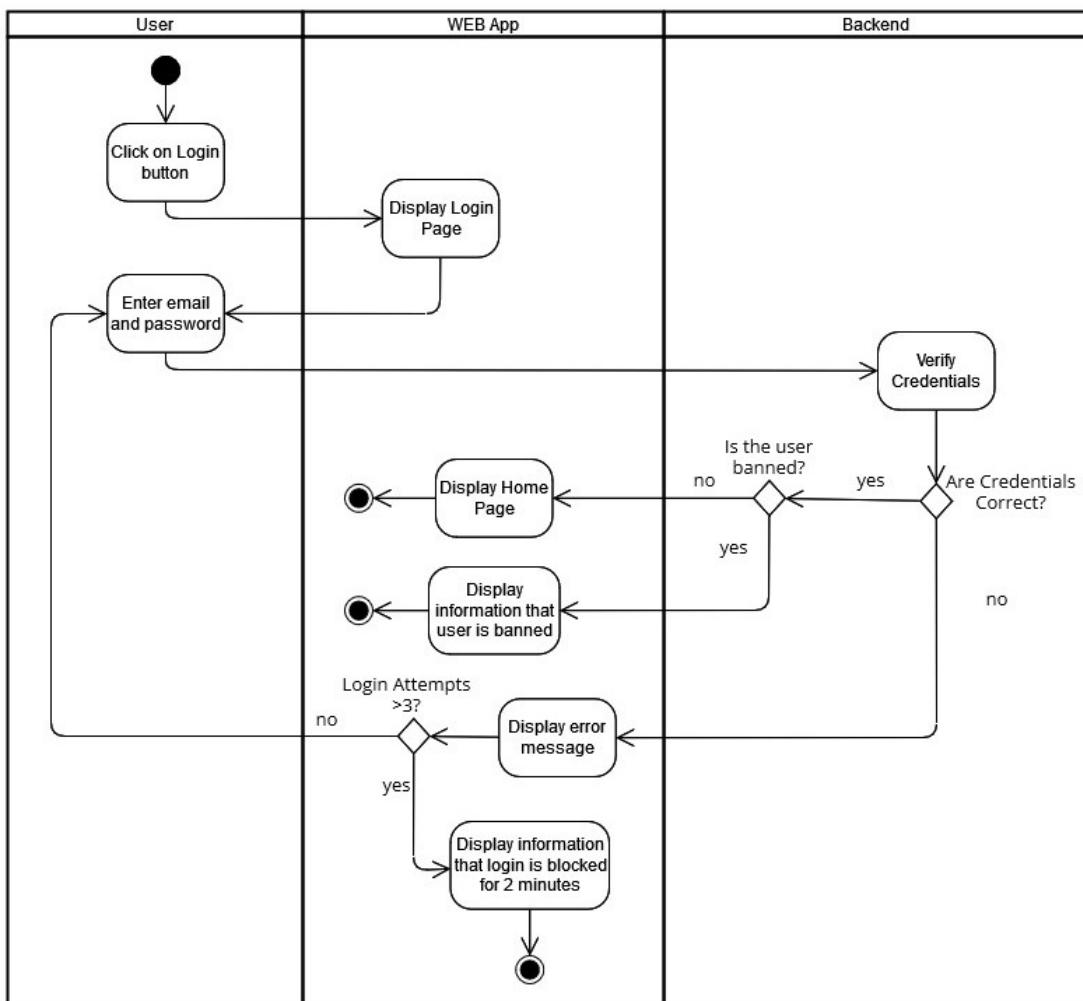


Fig. 16. Activity Diagram for Login Use Case

### New Session Process

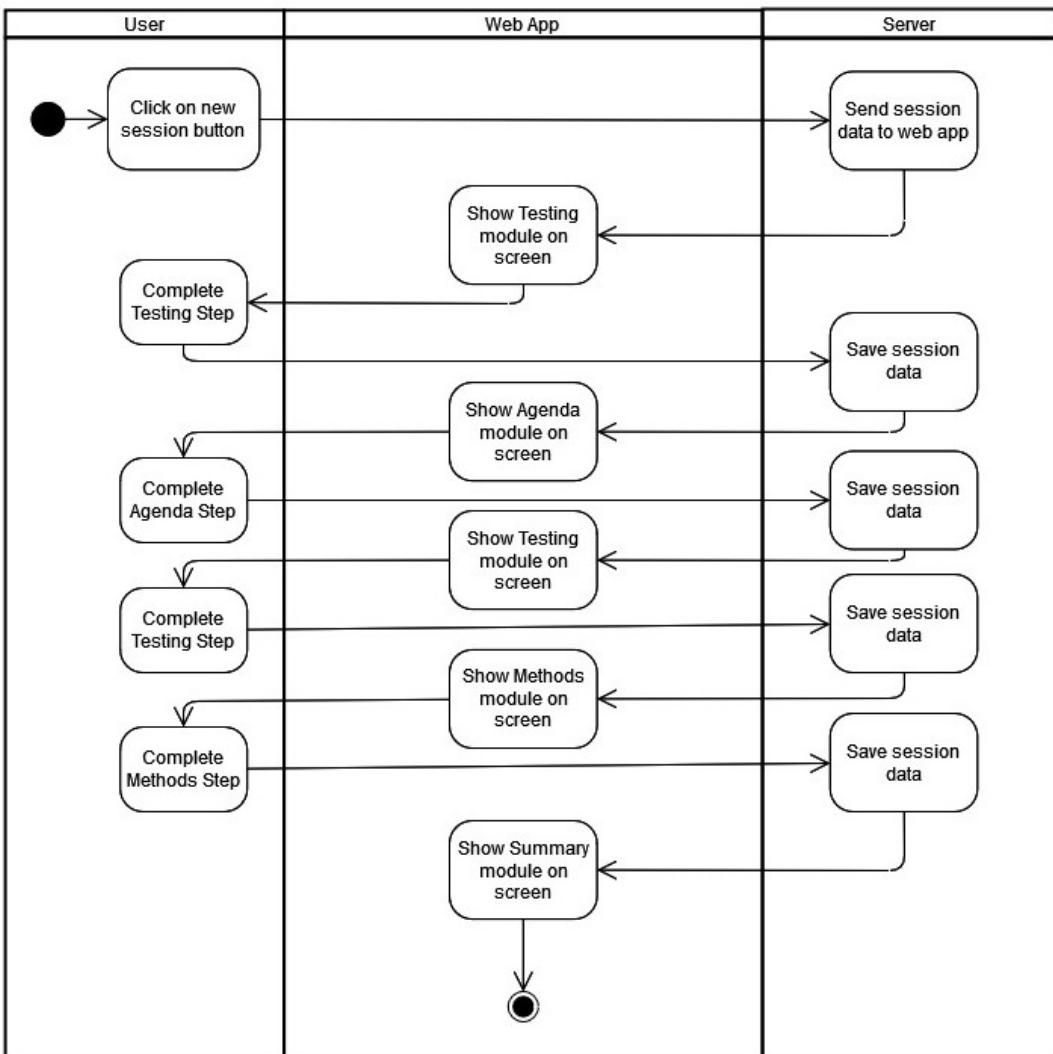


Fig. 17. Activity Diagram for New Session Use Case

### See Statistics

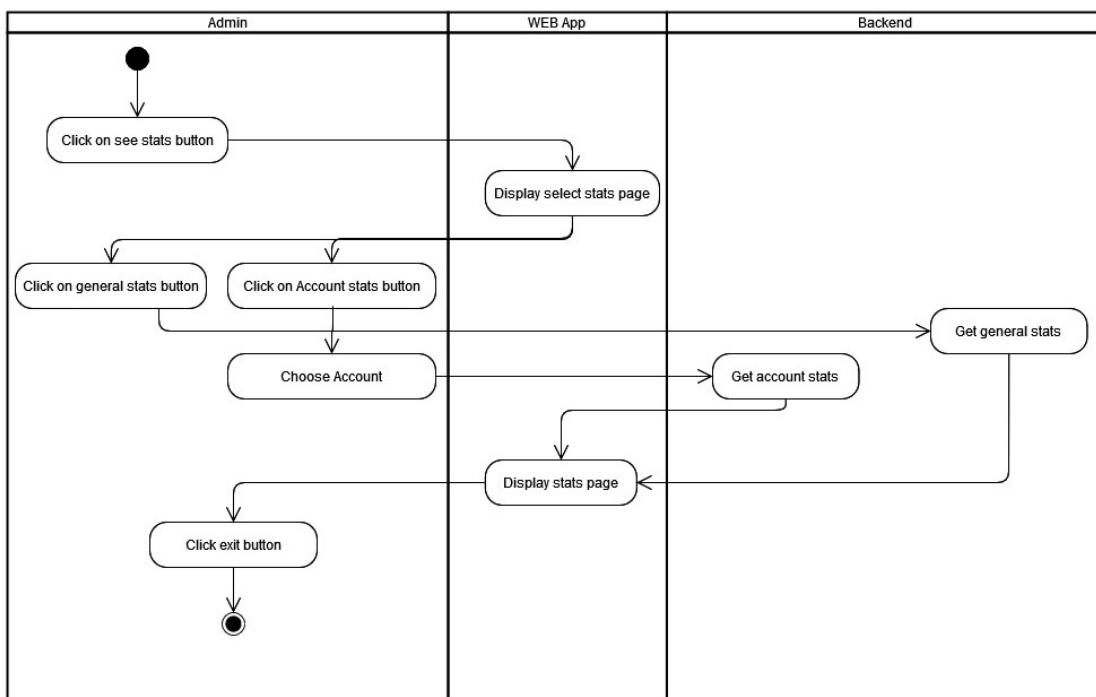


Fig. 18. Activity Diagram for See Stats Use Case

## Register

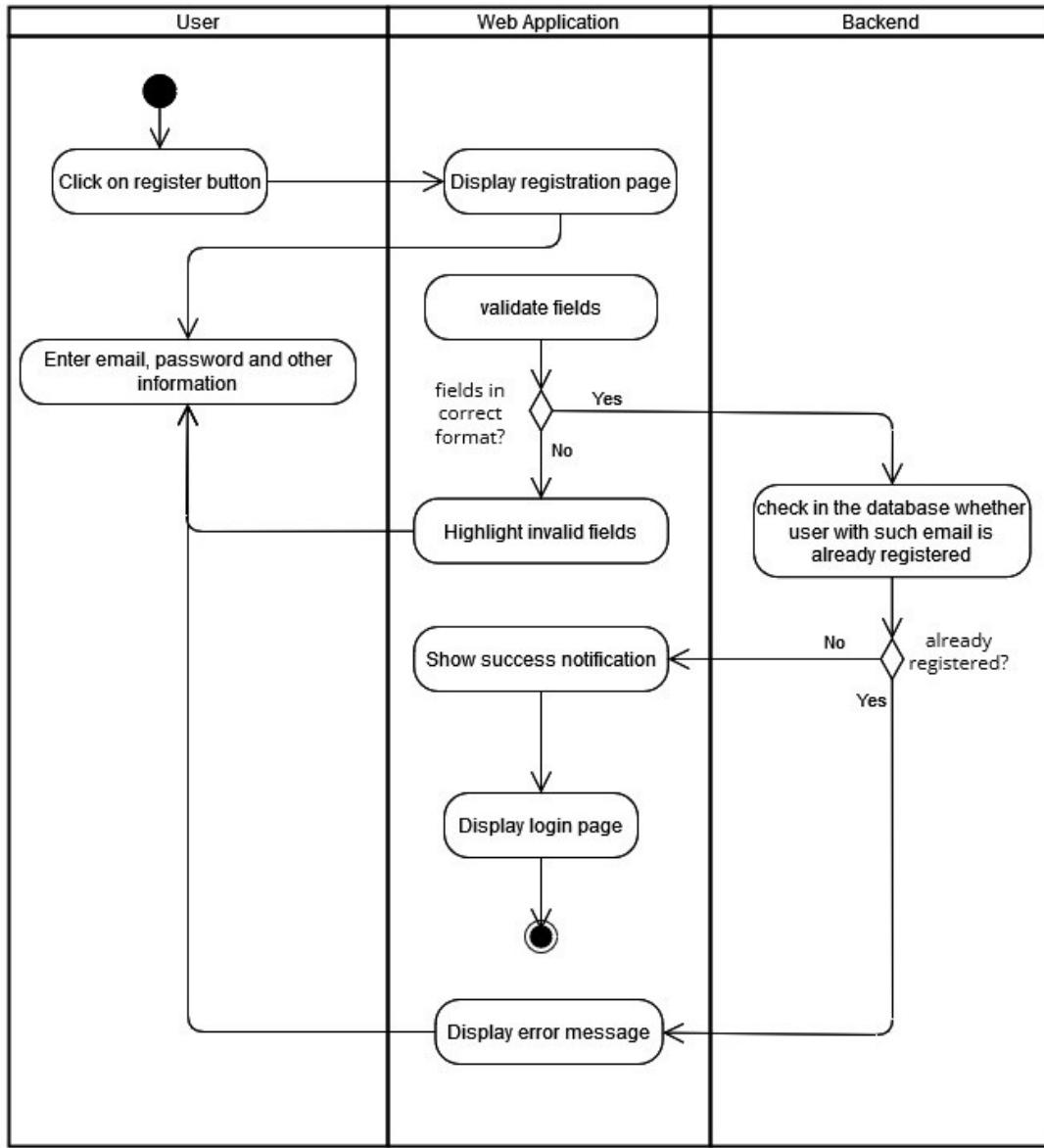


Fig. 19. Activity Diagram for Registration Use Case

## Evaluation

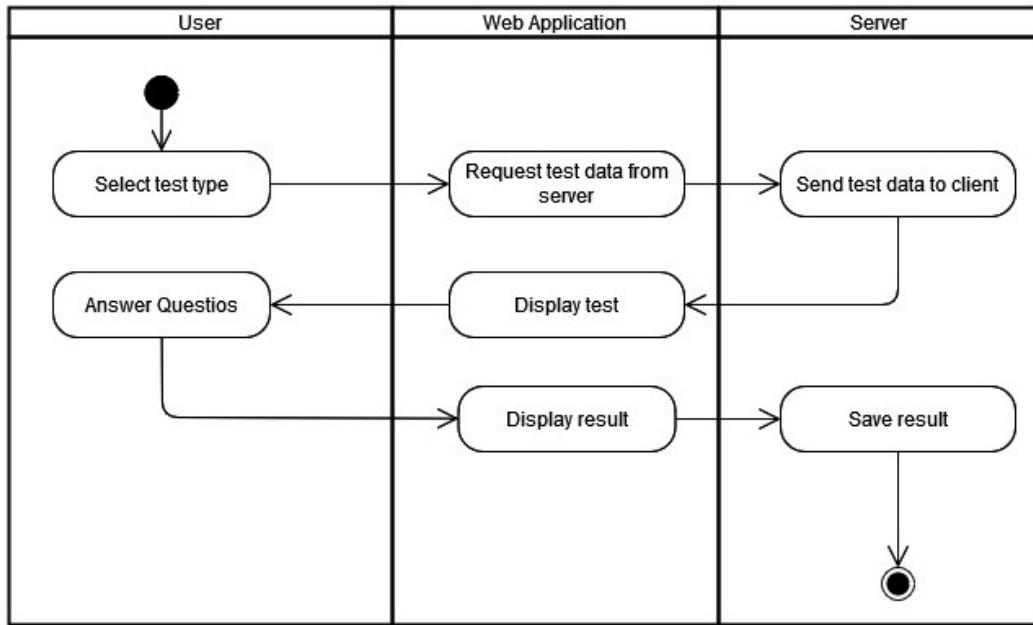


Fig. 20. Activity Diagram for Evaluation Use Case

Continue Previous Session

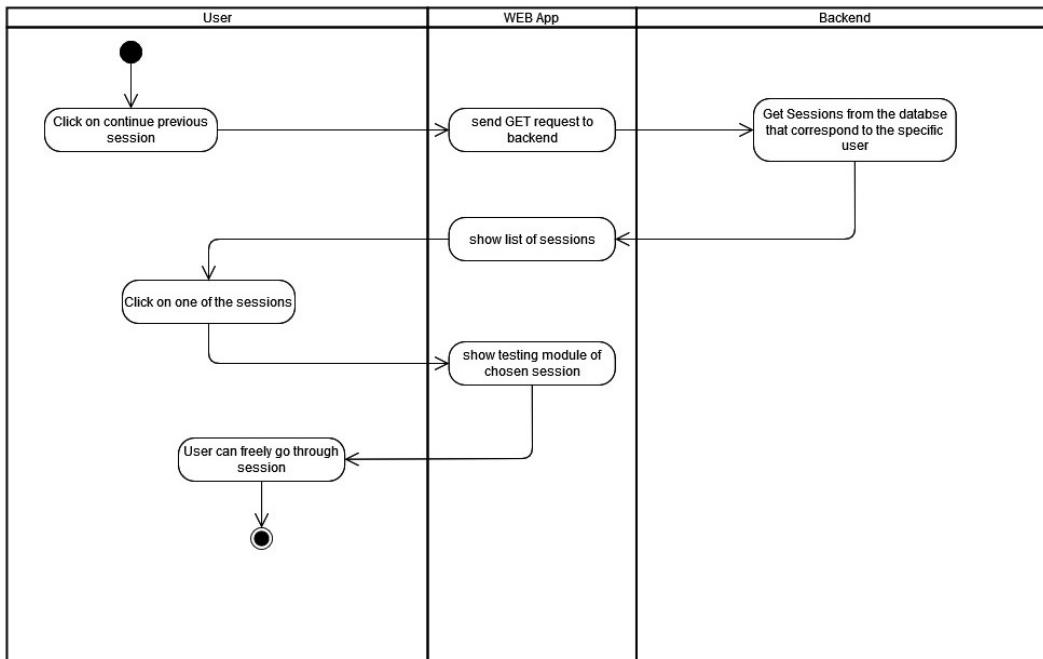


Fig. 21. Activity Diagram for Continue Previous Session Use Case

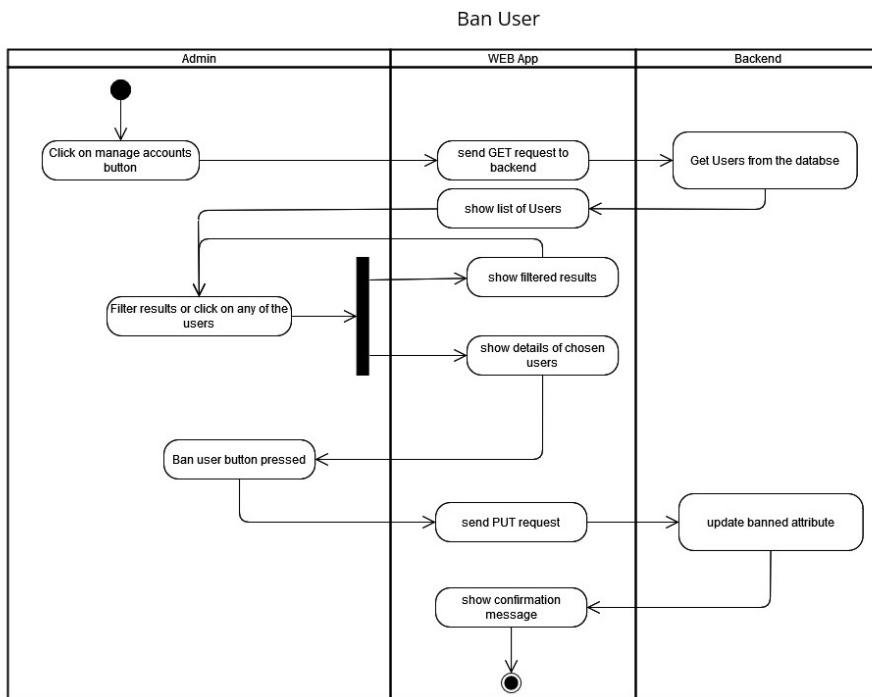


Fig. 22. Activity Diagram for Manage Accounts Use Case

### Manage account

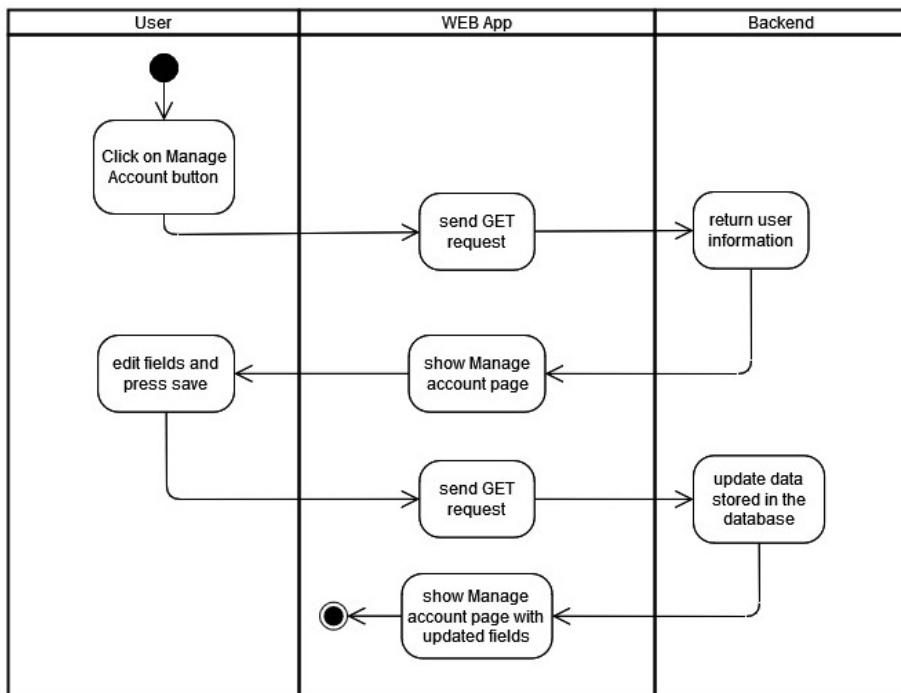


Fig. 23. Activity Diagram for Manage Account Use Case

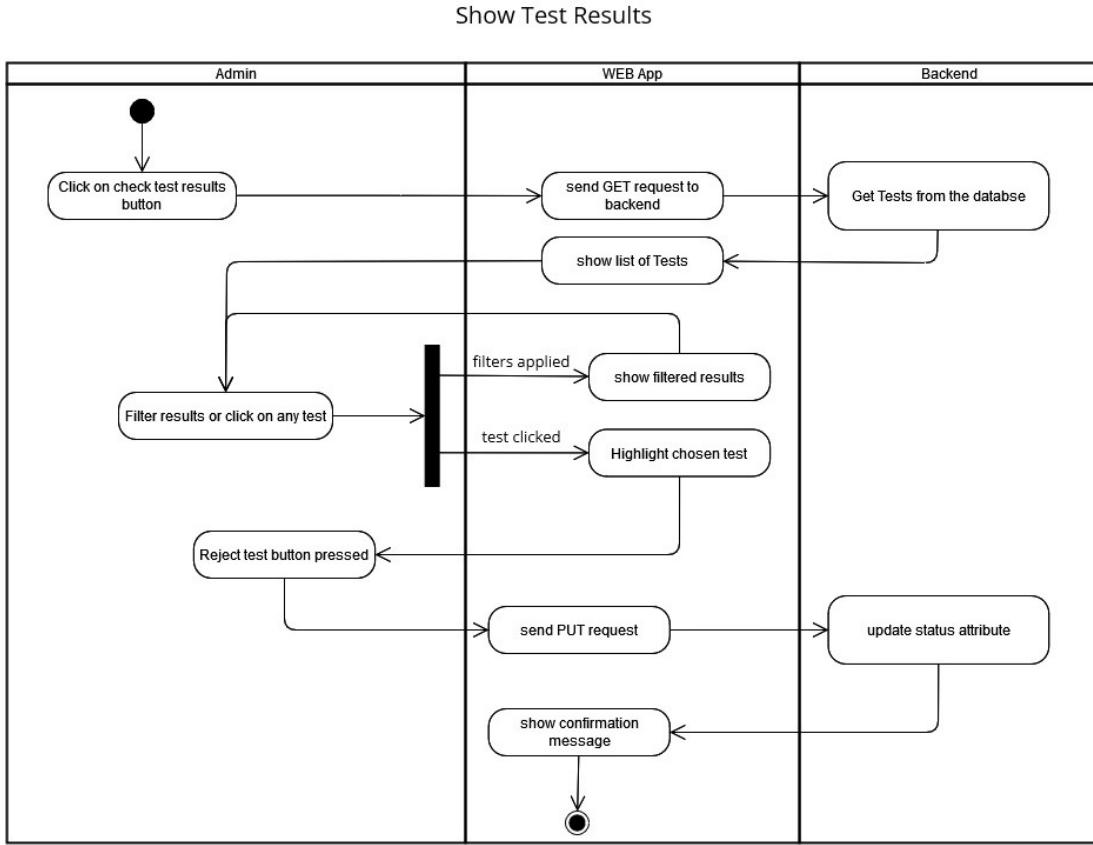


Fig. 24. Activity Diagram for Show Test Results Use Case

## 8. Communication Protocol

**8.1. General Design**—Communication between web application and the server will be facilitated using HTTP, with messages in JSON in the form provided in the below API. Data from the database will be retrieved by using queries with the help of database driver.

**8.2. Sequence Diagrams**—On each of the sequence diagrams communication between user/admin, web application, server and database is presented.

- Start a new session - The sequence diagram can be divided into stages that correspond to the steps of TEAM, user fills a given module (Testing, Agenda or Methods) and then it is verified, whether all information is filled in properly and after that it is saved in the database or in the case when something is off, an error message is displayed and the module needs to be corrected. After the last module, all the information is gathered and a summary is sent to the client.
- Continue previous session - User has the possibility to continue previous session, the most recently saved session, that has not been finished (it's state is not Finished) is fetched from the database by sessionID and is then sent to client.
- Login - User logins by entering proper email and password, the format is being validated at the web client, and then after pressing the login button, request is sent to the server and credentials are verified with those present at the database. In the

case of correct credentials, a success response is sent to frontend and homepage is displayed, when the credentials are incorrect, frontend receives one of the error codes, user is presented with an error message and has a chance to reenter credentials.

- Show Main Page - our system offers different interfaces for user and admin. A different page after login in should be displayed. Request is sent to the server to verify what type of user has just logged in. Backend checks whether userID corresponds to user or admin account. Depending on the response from the server a different page is displayed.
- Retrieving Users, Sessions or Tests - Since the system offers the functionality for user to be able to look at the list of previously completed sessions and for the admin to look at the list of users, we need a way to retrieve them from the database. A request is sent to the server that corresponds to the proper endpoint (presented in below API), a number of entries is retrieved from the database, we shouldn't be getting all the data at once (pagination), since that would slow the system down, by default the number of retrieved users/sessions is 20. If the operation was successful, frontend receives the data and presents it on the screen. The diagram has been made for retrieving users, however the process is analogous for sessions and tests.
- Register - Registration happens by entering email and password but also age and gender, the format of fields is validated on client side and then the registration request is validated at the server, it is checked in the database whether such user already exists. If the request is successful a new entry (row) is created in the database in the proper table (User table), an information that the operation succeeded is sent to frontend and then user receives a message that the registration has been done and is presented with a login page, otherwise an error message is shown and user can correct the data..
- See Statistics - After admin presses the see statistical data button a page with two other buttons is shown. Admin has a choice to see the overall statistics in each of the test categories, the number of points on average in each test over a period of time, a request is sent to the server, test data is retrieved from the database and then the statistics are prepared and displayed. Admin can also see personal data of a particular user, again the score in each category of test over a period of time.
- Start Evaluation - First test type needs to be selected from the following: Anxiety, Depression, Relationship, Addiction, Happiness next test data is retrieved from the database and the question set is shown to the user, the test is filled out, the results are saved in the database and finally the result with explanation is shown.
- Manage Account - Once user presses the manage account button, a page with fields containing user's data is shown, user edits the data and presses save, a *POST* request is sent to the server and the new information is saved in the database. If the operation was successful a page with new edited data is shown, otherwise user receives an error message.
- Ban User - Admin, after selecting desired users, presses ban user button, a request with a list of users to ban is sent to backend and the data is updated in the database, if the operation was successful, all the users now have the *banned* attribute set to true and information about the operation being completed is shown to the admin, otherwise an error message is shown.

- Filtering - Our system in some places offers data filtering, users can be filtered by *userID* or *email* and sessions can be filtered by *status*. One sequence diagram has been prepared, however they are analogous for other cases. A *GET* request is sent to a proper endpoint and backend retrieves the data from the database. Frontend receives the data in case of success and an error message in case of failure.
- Show Test Results - After pressing show test results button, frontend sends *GET* request to backend and tests are retrieved from the database. On successful operation Admin is presented with a list of tests, if the operation was a failure an error message is shown on the web app.

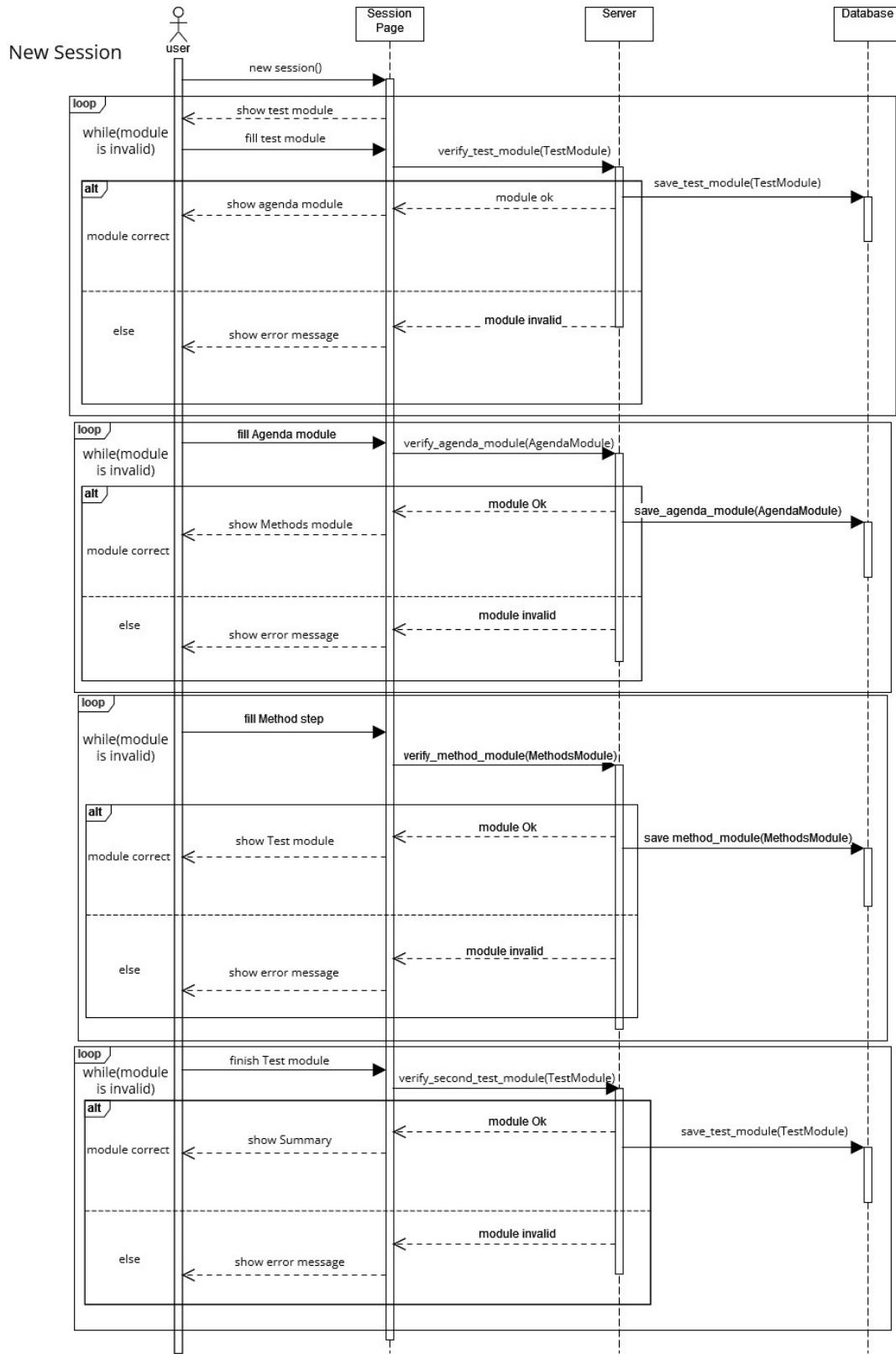


Fig. 25. Sequence Diagram for New Session Use Case

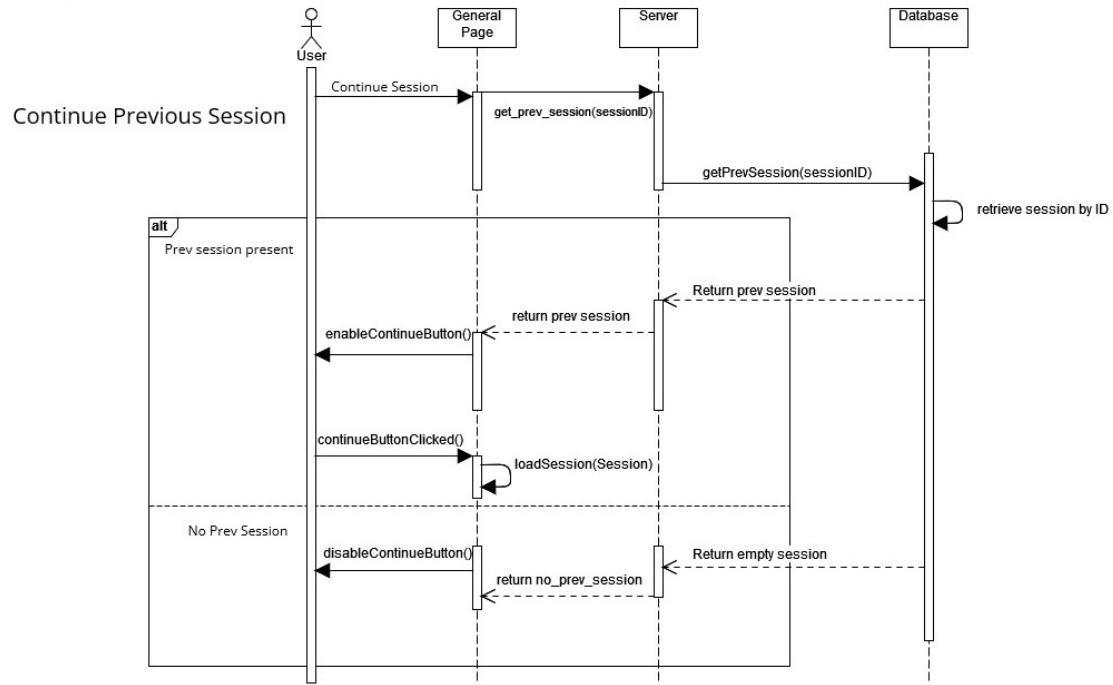
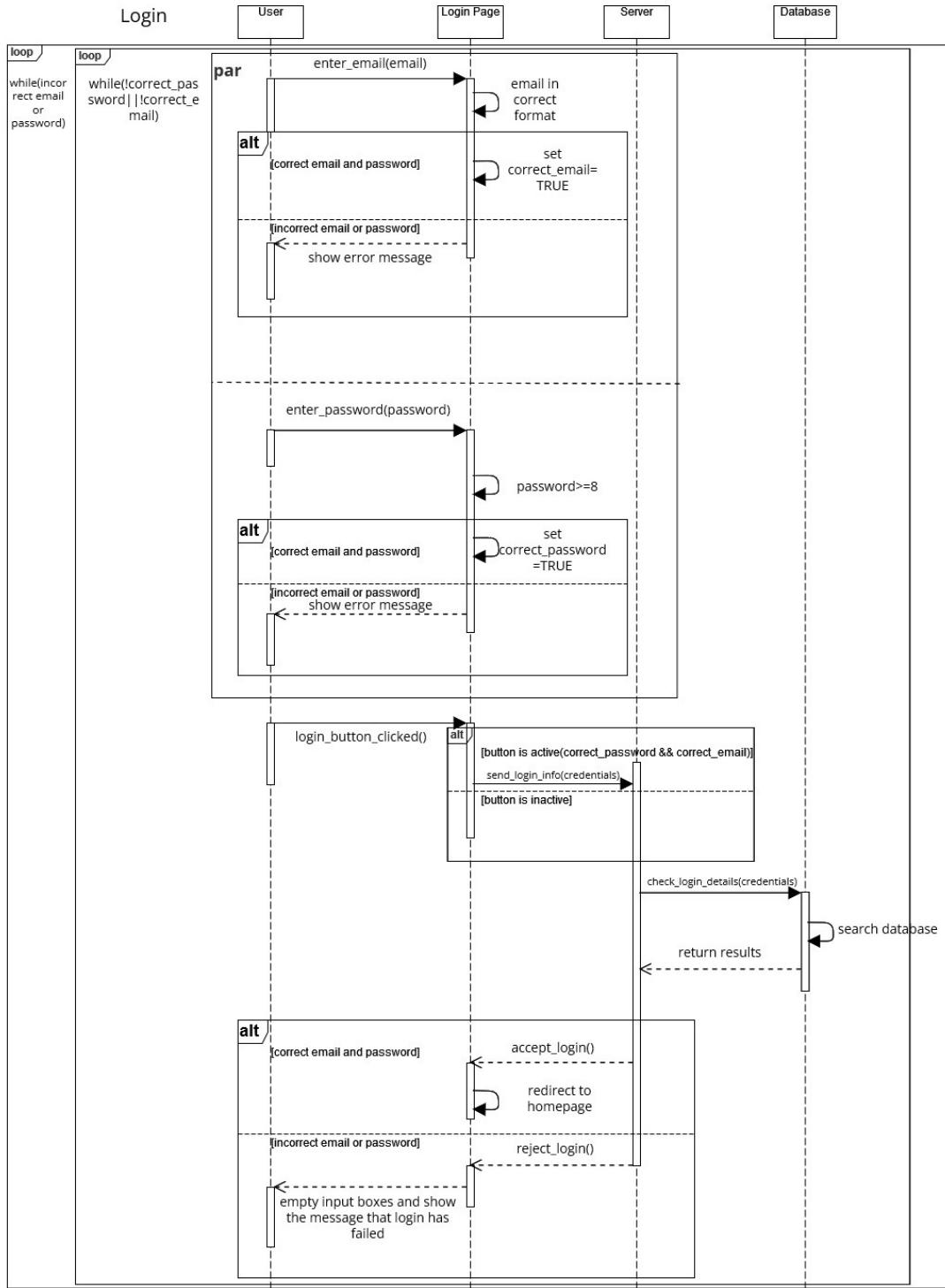


Fig. 26. Sequence Diagram for Continue Previous Session Use Case



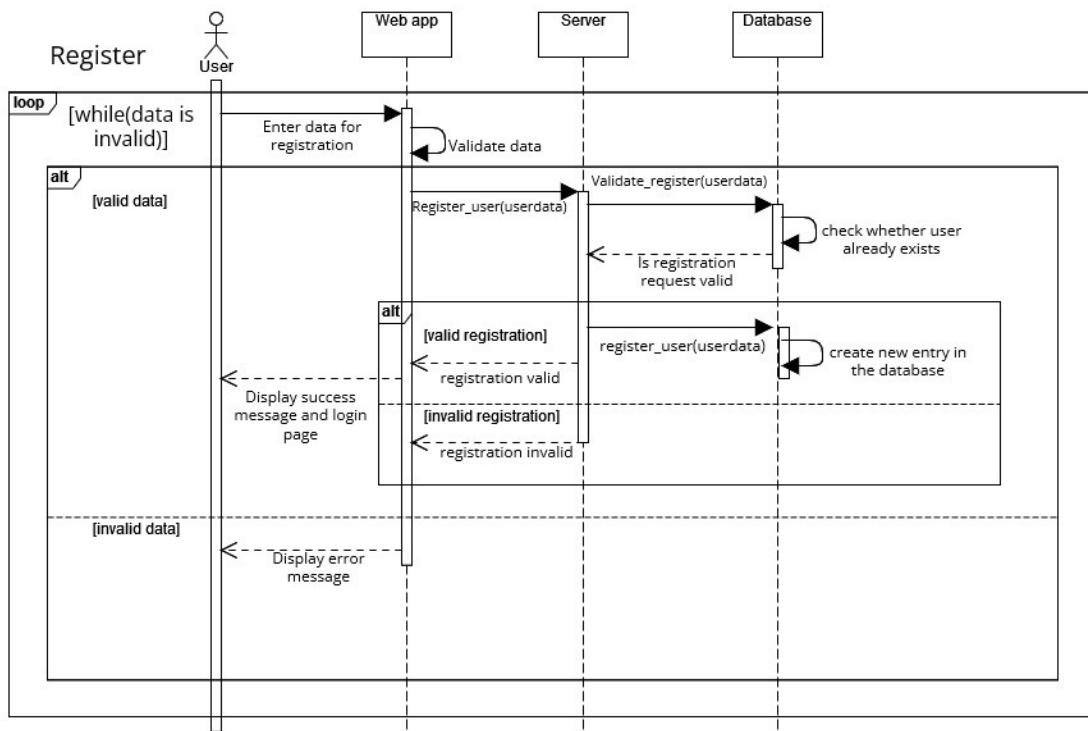


Fig. 28. Sequence Diagram for Register Diagram

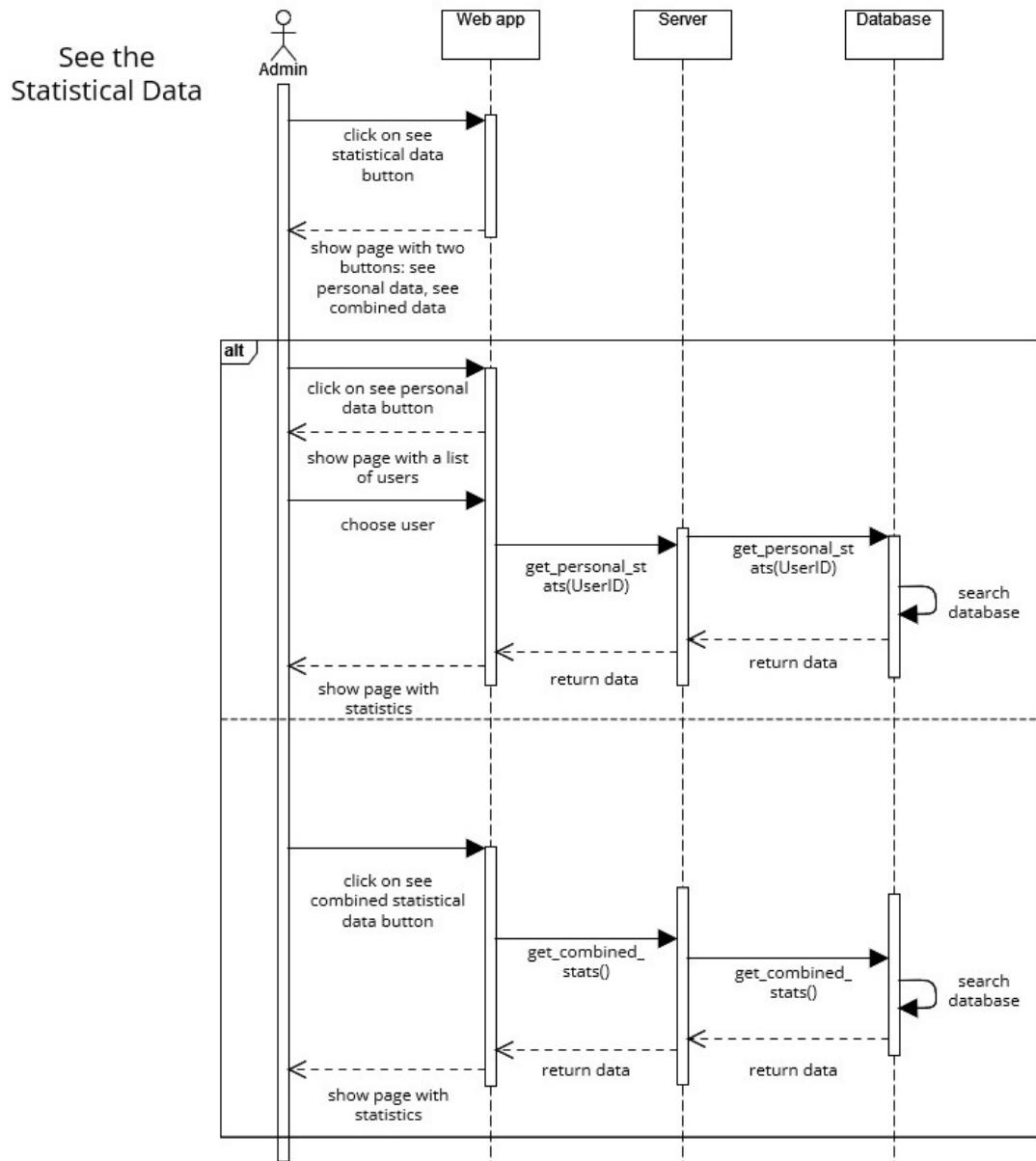


Fig. 29. Sequence Diagram for See Statistics Use Case

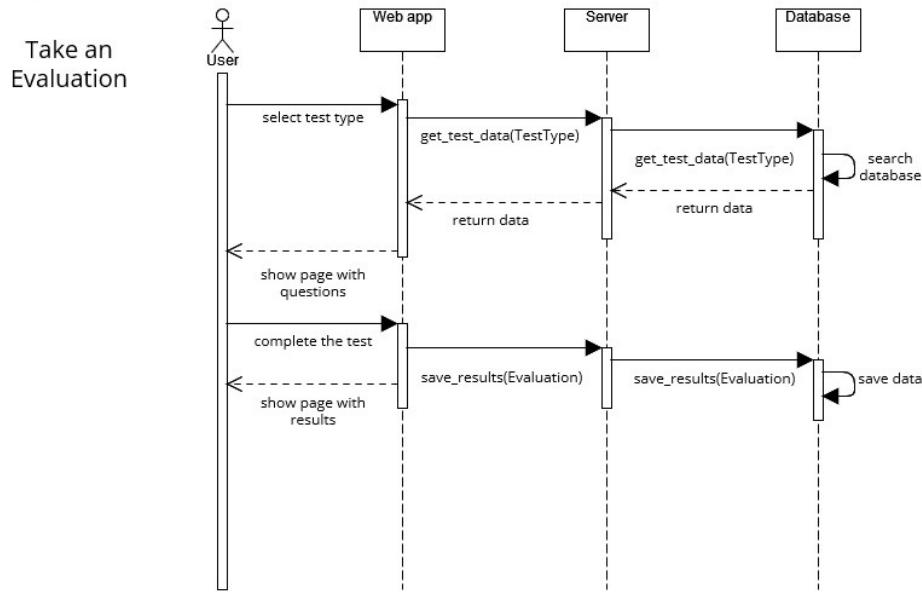


Fig. 30. Sequence Diagram for Start Evaluation Use Case

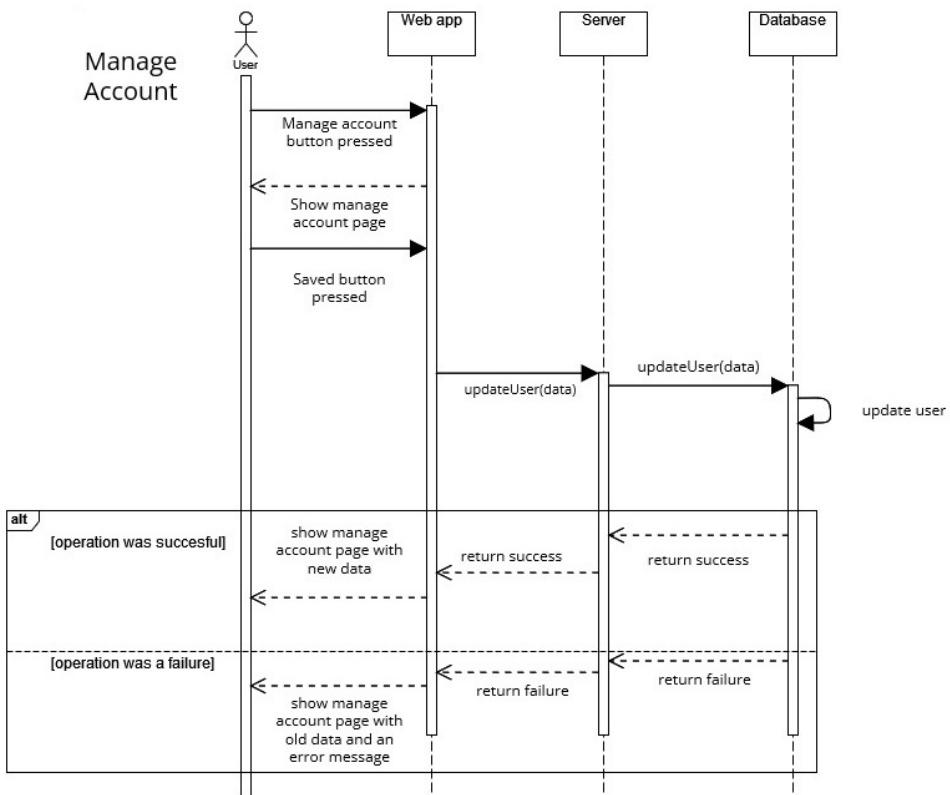


Fig. 31. Sequence Diagram for Manage Account Use Case

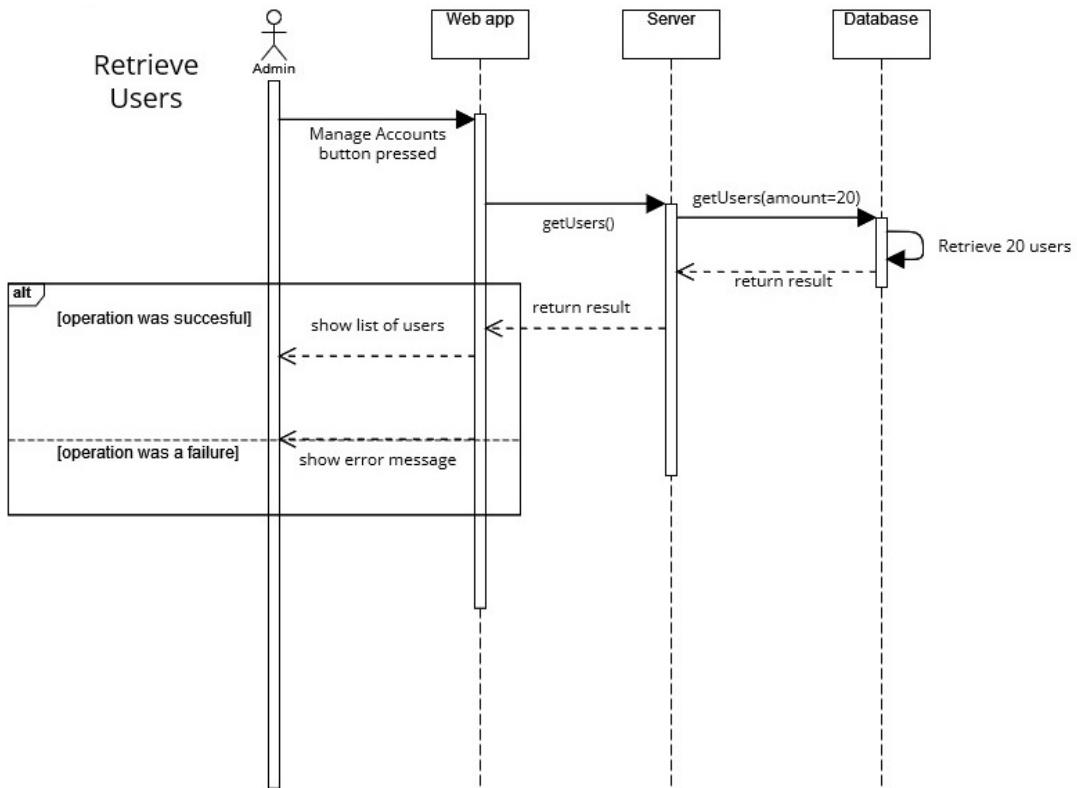


Fig. 32. Sequence Diagram for User Retrieval

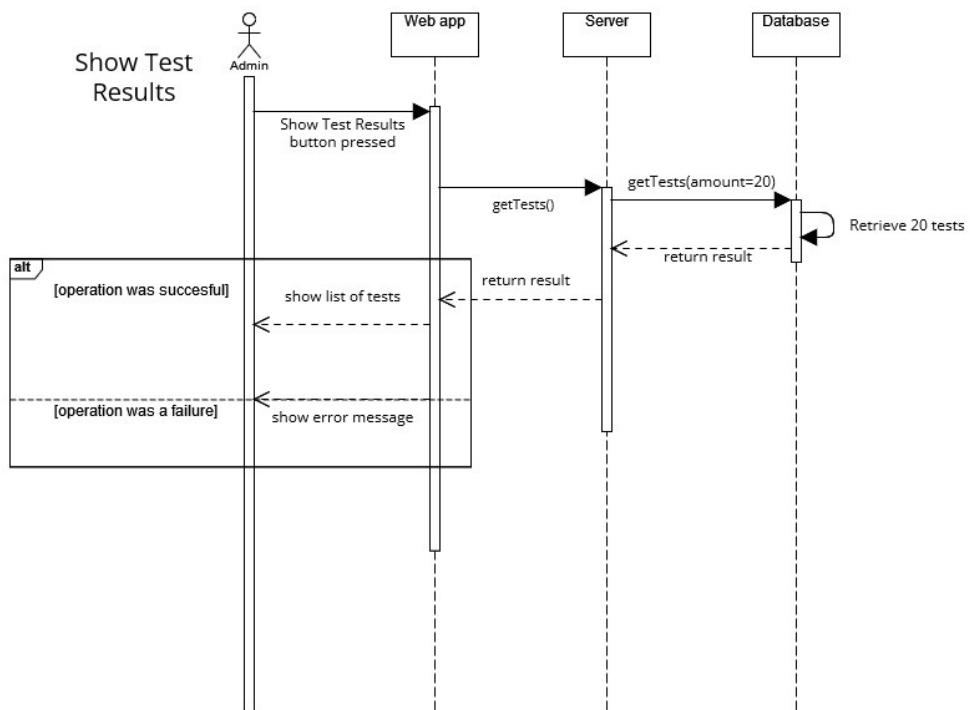


Fig. 33. Sequence Diagram for Show Test Results Use Case

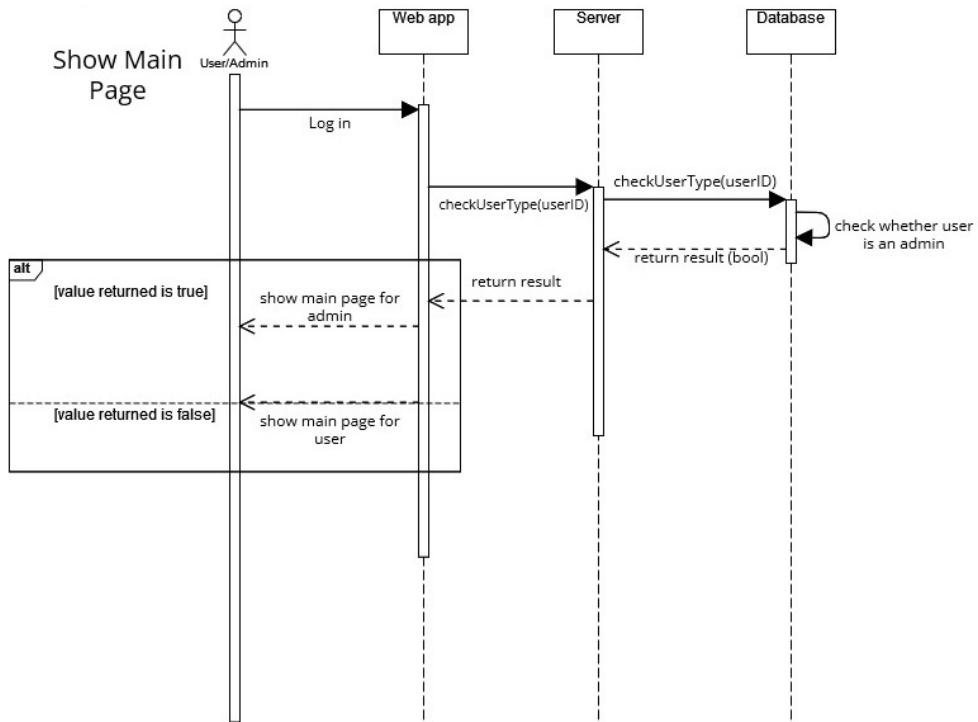


Fig. 34. Sequence Diagram for displaying Main Page

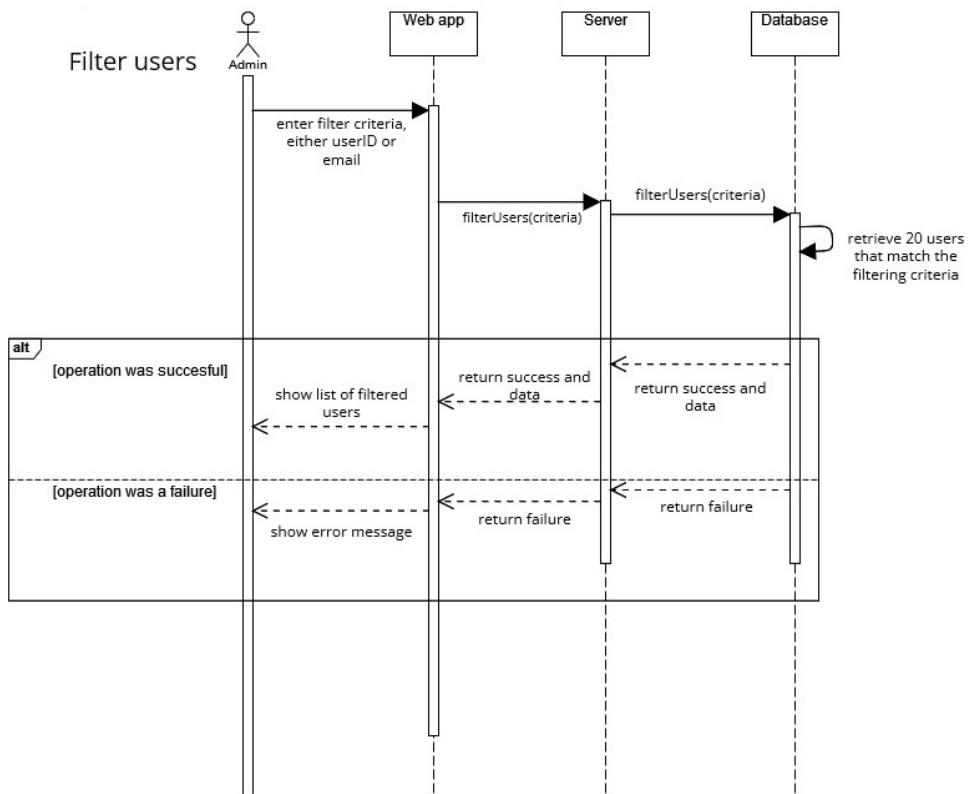


Fig. 35. Sequence Diagram for Filtering

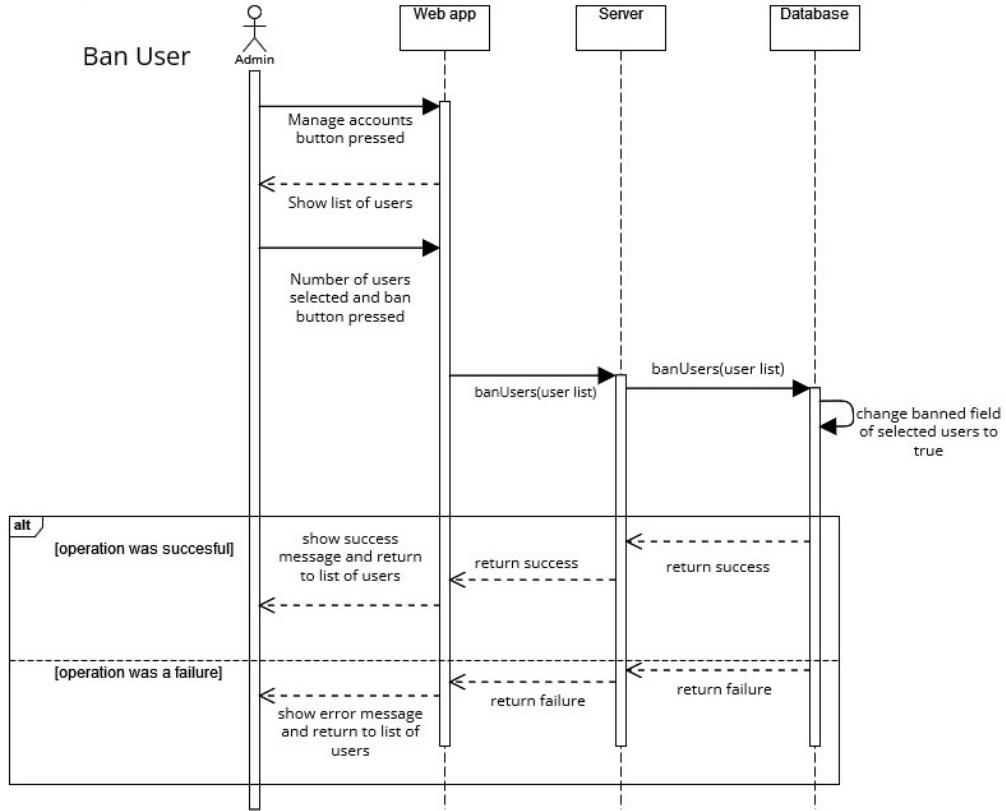


Fig. 36. Sequence Diagram for Banning Users

8.3. **API**—The API has been written in OpenAPI 3.0.1, which allows for automatic generation of server and client. Interactive description of the API is provided in separate API.html file, we have also attached javascript client and aspnet server generated from the API. There are 4 schemas present: User, Session, Evaluation and Statistics. For **User** we have 7 endpoints:

1. /user (POST for registering)
2. /user/ (GET to get a list of users)
3. /user/login (POST for logging in)
4. /user/logout (POST for logout)
5. /user/email (GET user by email)
6. /user/email (PUT for updating user, can also be used for banning)
7. /user/email (DELETE user)

For **session** there are 6 endpoints:

1. /session (POST to add a new session)
2. /session/findByID (GET find list of sessions that correspond to a particular user)
3. /session/findByStatus (GET find session by status)
4. /session/sessionID (GET find session by id)
5. /session/sessionID (POST to update a session)
6. /session/sessionID (DELETE to erase session data)

For **evaluation** we have defined 6 endpoints:

1. /evaluation/evaluationID (PUT to update an existing evaluation)

2. /evaluation (POST to add a new evaluation)
3. /evaluation/findByUserID (GET find list of evaluations by user's id)
4. /evaluation/findByStatus (GET find evaluation by status)
5. /evaluation/evaluationID (GET find evaluation by ID)
6. /evaluation/evaluationID (DELETE remove evaluation with a given ID)

For **statistics** 4 endpoints have been created:

1. /statistics/statisticID (PUT to update an existing statistic)
2. /statistics (POST to add a new statistic)
3. /statistics/email (GET to retrieve a statistic of a specific user)
4. /statistics/userID (GET to retrieve a list statistics of a specific user)

Schematics have been created for User, Session, Evaluation and Statistics and are displayed at the end of the below code listing.

```

1 openapi: 3.0.1
2 info:
3   title: Team-CBT API
4   description: 'This is an API for Team-CBT web app'
5   version: 1.0.0
6 servers:
7 - url: https://teamcbt.com/v1
8 - url: http://teamcbt.com/v1
9 tags:
10 - name: user
11   description: Operations about user
12 - name: session
13   description: Operations about session
14 - name: evaluation
15   description: Operations about evaluation
16 - name: statistics
17   description: Operations about statistics
18
19 paths:
20   /session:
21     post:
22       tags:
23         - session
24         summary: Add a new session
25         operationId: addSession
26         requestBody:
27           description: Session object that needs to be added
28           content:
29             application/json:
30               schema:
31                 $ref: '#/components/schemas/Session'
32             application/xml:
33               schema:
34                 $ref: '#/components/schemas/Session'
35             required: true
36         responses:
37           200:
38             description: Success
39             content: {}
40           400:
41             description: Invalid input
42             content: {}
43           403:
44             description: Access Forbidden

```

```

45     content: {}
46
47     503:
48         description: Server is currently down
49         content: {}
50
51     security:
52         - teamcbt_auth:
53             - write:session
54             - read:session
55
56     x-codegen-request-body-name: body
57
58 /session/findByStatus:
59
60     get:
61
62         tags:
63             - session
64
65         summary: Finds Session by status
66         description: Multiple status values can be provided with comma separated
67         strings
68
69         operationId: findSessionByStatus
70
71         parameters:
72             - name: status
73                 in: query
74                 description: Status values that need to be considered for filter
75                 required: true
76
77             style: form
78             explode: true
79
80             schema:
81                 type: array
82                 items:
83                     type: string
84                     default: finished
85
86                     enum:
87                         - not_started
88                         - started_testing
89                         - started_agenda
90                         - started_methods
91                         - started_testing_finish
92                         - finished
93
94             - name: email
95                 in: query
96                 description: User's email
97                 required: false
98
99                 style: form
100                explode: true
101
102                schema:
103                    type: string
104
105        responses:
106
107            200:
108
109                description: successful operation
110                content:
111
112                    application/xml:
113                        schema:
114                            type: array
115                            items:
116                                $ref: '#/components/schemas/Session'
117
118                    application/json:
119                        schema:
120                            type: array
121                            items:
122                                $ref: '#/components/schemas/Session'
123
124            400:
125
126                description: Invalid status value or email
127                content: {}

```

```

105      403:
106          description: Access Forbidden
107          content: {}
108      503:
109          description: Server is currently down
110          content: {}
111      security:
112          - teamcbt_auth:
113              - write:session
114              - read:session
115  /session/findByUserID:
116      get:
117          tags:
118              - session
119          summary: Finds Session by UserID
120          operationId: findSessionByUserID
121          parameters:
122              - name: UserID
123                  in: query
124                  description: Status values that need to be considered for filter
125                  required: true
126                  style: form
127                  explode: true
128                  schema:
129                      type: integer
130          responses:
131              200:
132                  description: successful operation
133                  content:
134                      application/xml:
135                          schema:
136                              type: array
137                              items:
138                                  $ref: '#/components/schemas/Session'
139                      application/json:
140                          schema:
141                              type: array
142                              items:
143                                  $ref: '#/components/schemas/Session'
144              400:
145                  description: Invalid status value or email
146                  content: {}
147              403:
148                  description: Access Forbidden
149                  content: {}
150              503:
151                  description: Server is currently down
152                  content: {}
153      security:
154          - teamcbt_auth:
155              - write:session
156              - read:session
157  /session/{sessionID}:
158      get:
159          tags:
160              - session
161          summary: Find Session by ID
162          description: Returns a single session
163          operationId: getSessionById
164          parameters:
165              - name: sessionID

```

```

166     in: path
167     description: ID of session to return
168     required: true
169     schema:
170       type: integer
171       format: int64
172   responses:
173     200:
174       description: successful operation
175       content:
176         application/xml:
177           schema:
178             $ref: '#/components/schemas/Session'
179         application/json:
180           schema:
181             $ref: '#/components/schemas/Session'
182     400:
183       description: Invalid ID supplied
184       content: {}
185     403:
186       description: Access Forbidden
187       content: {}
188     404:
189       description: Session not found
190       content: {}
191     503:
192       description: Server is currently down
193       content: {}
194   security:
195     - api_key: []
196   put:
197     tags:
198     - session
199     summary: Updates a Session in the database with form data
200     operationId: updateSessionWithForm
201     parameters:
202     - name: sessionId
203       in: path
204       description: ID of session that needs to be updated
205       required: true
206       schema:
207         type: integer
208         format: int64
209     requestBody:
210       content:
211         '*/*':
212           schema:
213             $ref: '#/components/schemas/Session'
214             required: true
215   responses:
216     405:
217       description: Invalid input
218       content: {}
219   security:
220     - teamcbt_auth:
221       - write:session
222       - read:session
223   delete:
224     tags:
225     - session
226     summary: Deletes a session

```

```

227     operationId: deleteSession
228     parameters:
229     - name: api_key
230         in: header
231         schema:
232             type: string
233     - name: sessionID
234         in: path
235         description: Session id to delete
236         required: true
237         schema:
238             type: integer
239             format: int64
240     responses:
241     400:
242         description: Invalid ID supplied
243         content: {}
244     403:
245         description: Access Forbidden
246         content: {}
247     404:
248         description: Session not found
249         content: {}
250     503:
251         description: Server is currently down
252         content: {}
253     security:
254     - teamcbt_auth:
255         - write:session
256         - read:session
257 /user:
258     post:
259         tags:
260         - user
261         summary: Register user
262         description: Register user
263         operationId: createUser
264         requestBody:
265             description: Created user object
266             content:
267                 '*/*':
268                 schema:
269                     $ref: '#/components/schemas/User'
270             required: true
271     responses:
272     200:
273         description: successful operation
274         content: {}
275     400:
276         description: Operation failed or User already exists
277         content: {}
278     403:
279         description: Access Forbidden
280         content: {}
281     503:
282         description: Server is currently down
283         content: {}
284     x-codegen-request-body-name: body
285     security:
286     - teamcbt_auth:
287         - write:user

```

```

288     get:
289         tags:
290         - user
291         summary: get list of users
292         operationId: getAllUsers
293         responses:
294             200:
295                 description: successful operation
296                 content:
297                     application/xml:
298                         schema:
299                             type: array
300                             items:
301                                 $ref: '#/components/schemas/User'
302                     application/json:
303                         schema:
304                             type: array
305                             items:
306                                 $ref: '#/components/schemas/User'
307             403:
308                 description: Access Forbidden
309                 content: {}
310             500:
311                 description: Failure of users retrieval
312                 content: {}
313             503:
314                 description: Server is currently down
315                 content: {}
316         security:
317         - teamcbt_auth:
318             - write:session
319             - read:session
320     /user/login:
321         post:
322             tags:
323             - user
324             summary: Logs user into the system
325             operationId: loginUser
326             parameters:
327             - name: email
328                 in: query
329                 description: Email of the user
330                 required: true
331                 schema:
332                     type: string
333             - name: password
334                 in: query
335                 description: The password for login in clear text
336                 required: true
337                 schema:
338                     type: string
339             responses:
340                 200:
341                     description: successful operation
342                     headers:
343                         X-Expires-After:
344                             description: date in UTC when token expires
345                             schema:
346                                 type: string
347                                 format: date-time
348                     content:

```

```

349         application/xml:
350             schema:
351                 type: string
352         application/json:
353             schema:
354                 type: string
355     400:
356         description: Invalid username/password supplied
357         content: {}
358     403:
359         description: Access Forbidden
360         content: {}
361     503:
362         description: Server is currently down
363         content: {}
364     security:
365     - teamcbt_auth:
366         - read:user
367 /user/logout:
368     post:
369         tags:
370         - user
371         summary: Logs out current logged in user session
372         operationId: logoutUser
373         responses:
374             200:
375                 description: successful operation
376                 content: {}
377             403:
378                 description: Access Forbidden
379                 content: {}
380             500:
381                 description: failure
382                 content: {}
383             503:
384                 description: Server is currently down
385                 content: {}
386 /user/{email}:
387     get:
388         tags:
389         - user
390         summary: Get user by email
391         operationId: getUserByEmail
392         parameters:
393         - name: email
394             in: path
395             description: 'The email of the user that needs to be fetched. Use
user1@gmail.com for testing. '
396             required: true
397             schema:
398                 type: string
399         responses:
400             200:
401                 description: successful operation
402                 content:
403                     application/xml:
404                         schema:
405                             $ref: '#/components/schemas/User'
406                     application/json:
407                         schema:
408                             $ref: '#/components/schemas/User'

```

```

409         400:
410             description: Invalid email supplied
411             content: {}
412         403:
413             description: Access Forbidden
414             content: {}
415         404:
416             description: User not found
417             content: {}
418         503:
419             description: Server is currently down
420             content: {}
421     put:
422         tags:
423             - user
424         summary: Updated user
425         description: This can be used for manage account and ban user use cases
426         operationId: updateUser
427         parameters:
428             - name: email
429                 in: path
430                 description: name that need to be updated
431                 required: true
432             schema:
433                 type: string
434         requestBody:
435             description: Updated user object
436             content:
437                 '*/*':
438                     schema:
439                         $ref: '#/components/schemas/User'
440                         required: true
441         responses:
442             400:
443                 description: Invalid user supplied
444                 content: {}
445             403:
446                 description: Access Forbidden
447                 content: {}
448             404:
449                 description: User not found
450                 content: {}
451             500:
452                 description: failure
453                 content: {}
454             503:
455                 description: Server is currently down
456                 content: {}
457         x-codegen-request-body-name: body
458     delete:
459         tags:
460             - user
461         summary: Delete user
462         description: This can only be done by the logged in user.
463         operationId: deleteUser
464         parameters:
465             - name: email
466                 in: path
467                 description: The email that needs to be deleted
468                 required: true
469             schema:

```

```

470         type: string
471     responses:
472         400:
473             description: Invalid username supplied
474             content: {}
475         403:
476             description: Access Forbidden
477             content: {}
478         404:
479             description: User not found
480             content: {}
481         500:
482             description: failure
483             content: {}
484         503:
485             description: Server is currently down
486             content: {}
487     /evaluation:
488         post:
489             tags:
490             - evaluation
491             summary: Add a new evaluation
492             operationId: addEvaluation
493             requestBody:
494                 description: Evaluation object that needs to be added
495                 content:
496                     application/json:
497                         schema:
498                             $ref: '#/components/schemas/Evaluation'
499                     application/xml:
500                         schema:
501                             $ref: '#/components/schemas/Evaluation'
502                         required: true
503             responses:
504                 200:
505                     description: Success
506                     content: {}
507                 400:
508                     description: Invalid input
509                     content: {}
510                 403:
511                     description: Access Forbidden
512                     content: {}
513                 500:
514                     description: failure
515                     content: {}
516                 503:
517                     description: Server is currently down
518                     content: {}
519             security:
520             - teamcbt_auth:
521                 - write:evaluation
522                 - read:evaluation
523             x-codegen-request-body-name: body
524     /evaluation/findByStatus:
525         get:
526             tags:
527             - evaluation
528             summary: Finds Evaluation by status
529             description: Multiple status values can be provided with comma separated

```

```

530     operationId: findEvaluationByStatus
531     parameters:
532       - name: status
533         in: query
534         description: Status values that need to be considered for filter
535         required: true
536         style: form
537         explode: true
538         schema:
539           type: array
540           items:
541             type: string
542             default: Not_Started
543             enum:
544               - Not_Started
545               - In_Progress
546               - On_Hold
547               - Finished
548       - name: email
549         in: query
550         description: User's email
551         required: false
552         style: form
553         explode: true
554         schema:
555           type: string
556     responses:
557       200:
558         description: successful operation
559         content:
560           application/xml:
561             schema:
562               type: array
563               items:
564                 $ref: '#/components/schemas/Evaluation'
565           application/json:
566             schema:
567               type: array
568               items:
569                 $ref: '#/components/schemas/Evaluation'
570       400:
571         description: Invalid status value or email
572         content: {}
573       403:
574         description: Access Forbidden
575         content: {}
576       500:
577         description: failure
578         content: {}
579       503:
580         description: Server is currently down
581         content: {}
582         security:
583           - teamcbt_auth:
584             - write:evaluation
585             - read:evaluation
586     /evaluation/findByID:
587       get:
588         tags:
589           - evaluation
590         summary: Finds Evaluation by userID

```

```

591     operationId: findEvaluationByuserID
592     parameters:
593     - name: userID
594         in: query
595         description: User's ID
596         required: true
597         style: form
598         explode: true
599         schema:
600             type: integer
601     responses:
602     200:
603         description: successful operation
604         content:
605             application/xml:
606                 schema:
607                     type: array
608                     items:
609                         $ref: '#/components/schemas/Evaluation'
610             application/json:
611                 schema:
612                     type: array
613                     items:
614                         $ref: '#/components/schemas/Evaluation'
615     400:
616         description: Invalid status value or email
617         content: {}
618     403:
619         description: Access Forbidden
620         content: {}
621     500:
622         description: failure
623         content: {}
624     503:
625         description: Server is currently down
626         content: {}
627     security:
628     - teamcbt_auth:
629         - write:evaluation
630         - read:evaluation
631 /evaluation/{evaluationID}:
632     get:
633         tags:
634         - evaluation
635         summary: Find Evaluation by ID
636         description: Returns a single session
637         operationId: getEvaluationById
638         parameters:
639         - name: evaluationID
640             in: path
641             description: ID of evaluation to return
642             required: true
643             schema:
644                 type: integer
645                 format: int64
646         responses:
647         200:
648             description: successful operation
649             content:
650                 application/xml:
651                     schema:

```

```

652         $ref: '#/components/schemas/Evaluation'
653     application/json:
654         schema:
655             $ref: '#/components/schemas/Evaluation'
656     400:
657         description: Invalid ID supplied
658         content: {}
659     403:
660         description: Access Forbidden
661         content: {}
662     404:
663         description: Evaluation not found
664         content: {}
665     500:
666         description: failure
667         content: {}
668     503:
669         description: Server is currently down
670         content: {}
671     security:
672         - api_key: []
673     put:
674         tags:
675             - evaluation
676         summary: Updates an Evaluation in the database with form data
677         operationId: updateEvaluationWithForm
678         parameters:
679             - name: evaluationID
680                 in: path
681                 description: ID of evaluation that needs to be updated
682                 required: true
683                 schema:
684                     type: integer
685                     format: int64
686         requestBody:
687             content:
688                 '*/*':
689                     schema:
690                         $ref: '#/components/schemas/Evaluation'
691                         required: true
692             responses:
693                 400:
694                     description: Invalid input
695                     content: {}
696                 403:
697                     description: Access Forbidden
698                     content: {}
699                 503:
700                     description: Server is currently down
701                     content: {}
702             security:
703                 - teamcbt_auth:
704                     - write:evaluation
705                     - read:evaluation
706     delete:
707         tags:
708             - evaluation
709         summary: Deletes an evaluation
710         operationId: deleteEvaluation
711         parameters:
712             - name: api_key

```

```

713     in: header
714     schema:
715       type: string
716   - name: evaluationID
717     in: path
718     description: Evaluation id to delete
719     required: true
720     schema:
721       type: integer
722       format: int64
723   responses:
724     400:
725       description: Invalid ID supplied
726       content: {}
727     403:
728       description: Access Forbidden
729       content: {}
730     404:
731       description: Evaluation not found
732       content: {}
733     500:
734       description: failure
735       content: {}
736     503:
737       description: Server is currently down
738       content: {}
739   security:
740   - teamcbt_auth:
741     - write:evaluation
742     - read:evaluation
743 /statistics/{statisticID}:
744   put:
745     tags:
746     - statistics
747     summary: Update an existing statistic
748     operationId: updateStatistics
749     parameters:
750     - name: statisticID
751       in: path
752       description: ID of statistic that needs to be updated
753       required: true
754       schema:
755         type: integer
756         format: int64
757     requestBody:
758       description: Evaluation object that needs to be updated
759       content:
760         application/json:
761           schema:
762             $ref: '#/components/schemas/Statistics'
763         application/xml:
764           schema:
765             $ref: '#/components/schemas/Statistics'
766           required: true
767   responses:
768     200:
769       description: Success
770       content: {}
771     400:
772       description: Invalid input
773       content: {}

```

```

774     403:
775         description: Access Forbidden
776         content: {}
777     500:
778         description: Server error
779         content: {}
780     503:
781         description: Server is currently down
782         content: {}
783     security:
784     - teamcbt_auth:
785         - write:statistics
786         - read:statistics
787     x-codegen-request-body-name: body
788 /statistics:
789     post:
790         tags:
791         - statistics
792         summary: Add a new statistic
793         operationId: addStatistic
794         requestBody:
795             description: Statistic object that needs to be added
796             content:
797                 application/json:
798                     schema:
799                         $ref: '#/components/schemas/Statistics'
800             application/xml:
801                 schema:
802                     $ref: '#/components/schemas/Statistics'
803             required: true
804         responses:
805             200:
806                 description: Success
807                 content: {}
808             400:
809                 description: Invalid input
810                 content: {}
811             403:
812                 description: Access Forbidden
813                 content: {}
814             500:
815                 description: Server error
816                 content: {}
817             503:
818                 description: Server is currently down
819                 content: {}
820     security:
821     - teamcbt_auth:
822         - write:statistics
823         - read:statistics
824     x-codegen-request-body-name: body
825 /statistics/{email}:
826     get:
827         tags:
828         - statistics
829         summary: Get Statistics of a user
830         description: Returns a statistics object
831         operationId: getStatisticsByEmail
832         parameters:
833             - name: email
834             in: path

```

```

835     description: email of user to return
836     required: true
837     schema:
838       type: string
839   responses:
840     200:
841       description: successful operation
842       content:
843         application/xml:
844           schema:
845             $ref: '#/components/schemas/Statistics'
846         application/json:
847           schema:
848             $ref: '#/components/schemas/Statistics'
849     400:
850       description: Invalid email supplied
851       content: {}
852     403:
853       description: Access Forbidden
854       content: {}
855     404:
856       description: Evaluation not found
857       content: {}
858     500:
859       description: Server error
860       content: {}
861     503:
862       description: Server is currently down
863       content: {}
864   security:
865     - teamcbt_auth:
866       - write:statistics
867       - read:statistics
868 /statistics/{userID}:
869   get:
870     tags:
871       - statistics
872     summary: Get Statistics of a user
873     description: Returns a statistics object
874     operationId: getStatisticsByUserID
875     parameters:
876       - name: userID
877         in: path
878         description: ID of user to return
879         required: true
880         schema:
881           type: integer
882     responses:
883       200:
884         description: successful operation
885         content:
886           application/xml:
887             schema:
888               $ref: '#/components/schemas/Statistics'
889         application/json:
890           schema:
891             $ref: '#/components/schemas/Statistics'
892     400:
893       description: Invalid ID supplied
894       content: {}
895     403:

```

```

896         description: Access Forbidden
897         content: {}
898     404:
899         description: Evaluation not found
900         content: {}
901     500:
902         description: Server error
903         content: {}
904     503:
905         description: Server is currently down
906         content: {}
907     security:
908     - teamcbt_auth:
909         - write:statistics
910         - read:statistics
911 components:
912 schemas:
913     User:
914         required:
915         - email
916         - password
917         - age
918         - gender
919         - banned
920         type: object
921         properties:
922             email:
923                 type: string
924             password:
925                 type: string
926             age:
927                 type: integer
928             gender:
929                 type: string
930             userStatus:
931                 type: integer
932             banned:
933                 type: boolean
934             xml:
935                 name: User
936     Session:
937         required:
938         - sessionID
939         - testStep
940         - agendaStep
941         - methodStep
942         type: object
943         properties:
944             sessionID:
945                 type: integer
946                 format: int64
947             testStep:
948                 type: object
949             agendaStep:
950                 type: object
951             methodStep:
952                 type: object
953             userID:
954                 type: integer
955             status:
956                 type: string

```

```

957         description: session status
958
959         enum:
960             - not_started
961             - started_testing
962             - started_agenda
963             - started_methods
964             - started_testing_finish
965             - finished
966
967     xml:
968         name: Session
969
970 Evaluation:
971     required:
972         - date
973         - status
974         - totalScore
975         - arrayScore
976         - questionSet
977         - evaluationID
978     type: object
979     properties:
980         evaluationID:
981             type: integer
982             format: int64
983         date:
984             type: string
985         status:
986             type: string
987             description: evaluation status
988             enum:
989                 - Not_Start
990                 - In_Progress
991                 - Finished
992                 - Rejected
993         totalScore:
994             type: integer
995         arrayScore:
996             type: array
997             items:
998                 type: integer
999         questionSet:
1000             type: array
1001             items:
1002                 type: string
1003
1004     xml:
1005         name: Evaluation
1006
1007 Statistics:
1008     required:
1009         - improvement
1010         - anxietyTestResults
1011         - depressionTestResults
1012         - relationshipTestResults
1013         - happinessTestResults
1014         - addictionTestResults
1015     type: object
1016     properties:
1017         statisticID:
1018             type: integer
1019         improvement:
1020             type: array
1021             items:
1022                 type: integer

```

```

1018     anxietyTestResults:
1019         type: array
1020         items:
1021             type: integer
1022     depressionTestResults:
1023         type: array
1024         items:
1025             type: integer
1026     relationshipTestResults:
1027         type: array
1028         items:
1029             type: integer
1030     happinessTestResults:
1031         type: array
1032         items:
1033             type: integer
1034     addictionTestResults:
1035         type: array
1036         items:
1037             type: integer
1038     date:
1039         type: string
1040     userID:
1041         type: integer
1042     xml:
1043         name: Statistics
1044     securitySchemes:
1045         teamcbt_auth:
1046             type: oauth2
1047             flows:
1048                 implicit:
1049                     authorizationUrl: http://teamcbt.com/oauth/dialog
1050             scopes:
1051                 write:session: modify session in your account
1052                 read:session: read your session
1053                 read:user: login to an account
1054                 write:evaluation: modify your evaluation
1055                 read:evaluation: read your evaluation
1056                 write:statistics: modify your statistics
1057                 read:statistics: read your statistics
1058     api_key:
1059         type: apiKey
1060         name: api_key
1061         in: header

```

## 9. User Simulations

In order to ensure that it will work properly in real time environment, the whole system needs to be tested, namely the presentation layer (frontend), business layer and database layer (backend). It can be done twofold: manually and automatically. Manual testes are performed by developers, whereas automatic tests are done by proper software, mentioned below. Testing will also reveal potential bugs, check the architecture's capability to handle traffic and perhaps create need for different solutions. There are many different tests that the system should go through, we have chosen those that are relevant for our project: Functionality, Usability, Interface, Database, Compatibility and Performance.

- Functionality Testing - this includes testing parameters such as user interface, APIs,

database testing, client and server testing and basic website functionalities. Each feature of the website should be tested. All links should lead to proper pages, every button should redirect to a desired site. All forms should check for mandatory fields, proper input formats, an error message should be shown if needed and once submitted, the data should be sent to the database. We would also like to test cookies, whether deleting cookies makes the website forget the user so that there is a need to enter credentials again. Most importantly tests should be performed in regard to end-to-end business scenarios, which take the user through a series of pages to complete, like the TEAM steps, including negative scenarios, such as when the user takes an unexpected action, for example wants to go to the next step without filling all required fields, an error message should be shown. **Selenium** is a great choice for functionality testing.

- Usability Testing - here we test the website's navigation including menus, buttons, links and we also check the content for spelling or grammatical errors and also for aesthetics, whether the pages are pleasing to the eye.
- Interface Testing - it should be checked whether requests to the database are sent properly and the output shown in web app is correct, in case of any errors, information about them should be shown to the administrator. Server should handle all requests sent by the web app without any service denial. All queries sent to the database return expected results. We should also test the system's response when the connection between the three layers cannot be established, then a message should be shown to the user. These can be done with the help of **Katalon**
- Database Testing - Database is a crucial component and so should be tested thoroughly, we should test whether any errors are shown when executing queries, data integrity is maintained while creating, updating or deleting data, check the performance and response times. **Selenium** comes to help again.
- Compatibility Testing - Same website will display differently based on which operating system and browser we are using. We should check for different combinations including operating systems: Windows, Linux, Mac, Android and IOS (mobile browsers) and browsers: Firefox, Chrome, Edge, Opera.
- Performance Testing - Our webapp needs to be tested under high loads of traffic and users. To do so we can use tools such as **Loadrunner** or **JMeter** taking into account various scenario's such as response time at different connection speeds, load test, stress test, and how fast the site recovers if it crashes.

For backend we can distinguish two types of tests - unit and integration tests. Unit tests check the functionality of isolated software components, for example specific class methods. Integration tests make sure that many components work properly together to produce a desired result. We can perform database tests such as:

- Structural testing - it involves tables validation, but also the validation of users, columns, keys, indexes, schemas, etc. Some helpful tools are: **SchemaCrawler** and **SP Test**.
- Functional testing - checks whether the frontend has the desired impact on the database, for example whether registering a new user, adds a new entry in the database.
- Non-functional tests - focuses on how the database performs under stress and also on the security. **HammerDB** can be helpful with that.

Many simulations of user actions can be defined, here, we cover all the essential scenarios:

1. Register a user account and check whether a new entry has been added to the database.
2. Log in with a newly created account, a main page with options to start a new session, continue previous session, take a mood test, manage account and logout should be displayed if the credentials were correct. It can also be checked whether the two minute blockade works correctly by trying to login with invalid credentials many times in a short time span.
3. Try to take a number of different mood tests: anxiety, depression, addiction, relationships, happiness and check whether the results are added to the database and then are correctly displayed for the user.
4. Start a new session: click new session button that sends a *GET* request to the server and the server returns the session data to frontend. User completes test module and sends a *PUT* request to the server, the server saves the data in the database and responds with the agenda module to the user. User completes agenda module and sends a *PUT* request to the server, the server saves the data in the database and responds with the Methods module to the user. Similarly it happens for the Methods module and the server at the end returns the final data, that is then displayed in a form of summary on the screen. It should be checked whether the data is saved in the database correctly, and the summary is consistent with what the user has written.
5. Press continue previous session button from the main screen, a list of previously started session should be displayed. A new session should be started before this test to check whether it will appear on the list. One of the list items should be chosen, then a *GET* request should be sent to backend and testing module of the chosen module of the chosen session should be displayed on the screen. User should be able to edit/fill fields and all the standard session functionality should be working properly.
6. Go to manage account page and try to change some information like password, a *PUT* request should be sent and new information should now appear in the database.
7. Press the log out button, a screen with options to log in, register and start a guest session should now appear.

We can also think of some testing scenarios for the administrator:

1. Some mood tests should be first filled in by users, and they should appear in the database with correct scores. After that the admin can press check test results button (other options are manage user accounts and see the statistical data) and should see a list of filled in tests, that can be filtered by time and also by *userID/email*. Filtering should display proper tests. Admin can reject any number of tests, in case it is decided that they have been completed by a malicious user or by a bot. Rejected tests should no longer appear on the list, when admin decides to check test results at some other time, and they should not be taken into account when computing statistics. These evaluations need not be removed, but their status should be changed to *Rejected* in *status* attribute. The rejections should be confirmed with the updated data in the database.
2. After pressing manage user accounts button, a list of users should be displayed, the list can be filtered out by *userID/email*. Admin can choose any of the users and decide to ban the user. After this user's *banned* field should be updated to true. This should

be confirmed with the database. The user should try to log in, an error message should be displayed and the user should not be able to log in.

3. Admin can view the statistical data, with a possibility to choose a particular user or to see the combined data of all users. If the choice is to see statistics of a particular user, then a list of users is displayed, users can be filtered out by *userID/email*. Admin can press on any user and then the statistics page with a graph is displayed. Admin can change the settings of the graph to show results of a particular test (depression, anxiety, addictions, relationships, happiness), also the time frame can be adjusted. By default on the graph there 5 lines, each representing different test, that concern time from the creation of user's account until now. If the user has not completed a single test from one of the categories, then that line should be missing from the graph. If admin chooses to see the combined statistical data, then he is presented with identical page as in the previous case, however this time the data doesn't concern just one user, but rather the whole user base. The correctness of graphs should be checked by taking a small amount of users, for example 10, each of them completing each test everyday for a week. The answers to the test don't have to be truthful, rather they shouldn't be to see whether the functions are displayed properly by the system. The results can be compared with those computed by hand.

Not all users/admins will behave in such a non-problematic way, oftentimes we will have to handle unexpected behaviour and so appropriate scenarios should be created, we describe some of them in abnormal behaviour section but we also leave that task to the imaginative minds of our colleagues.

## 10. Abnormal behaviour

It would be unreasonable to expect 100% uptime from all of the components. Some of them, whether it's the web app, server or database, will fail at times which may result in unexpected behaviour of the system. We need to be prepared to handle the cases when, for example, server goes down in the middle of the user going through TEAM session or perhaps the database takes unusually long to retrieve desired data.

We can think of situations when the user presses the button to start a session or an evaluation, request is sent to the server, but immediately after that the user closes the browser, the question then becomes whether we should delete the newly created object or rather keep it and give the user the possibility to continue to work on that new object once he logs in again. The solution we choose is different for Session and for an Evaluation. If the incident happens during the creation of session then it is saved in the database and can be continued next time user uses the web app, however if the same situation happens with an Evaluation, then it is also saved in the database with a score equal to 0, however it is then filtered out from the statistics, it is enough to check the state of an evaluation, if it is not "Finished" then it shouldn't be taken into consideration. We don't implement the possibility to continue a previous evaluation, simply because they take about 5 minutes to fill in, so that feature would be very rarely used and can be omitted.

The above example could be considered user's fault, but if during the session or an evaluation, web app loses connection with server? Ideally we would like to restore the

session from the data stored locally on user's side.

It might also happen that the database may become unresponsive. In such case we need to decide what information should be returned to the web app. We can simply return error code, that can be found in the API.

User should also be informed in case the saving process is faulty. After the completion of a certain module, the data should be saved in the database, it's crucial to let the user know in case the operation was unsuccessful, so that no further progress is made until the problem is resolved.

Some data should be kept locally, at least temporarily, so that when the user refreshes the page, the current progress is not lost and the filled in fields don't become blank.

Some people may have ill intentions and may want to try to guess other user's passwords by using brute-force attacks. We defend our system by limiting the number of failed attempts to login to 3 and then blocking that functionality for 2 minutes to prevent further attacks from a particular client.

There are many more abnormal situations that we just couldn't think of as of right now, they will appear during the development and testing, which is just a natural process of software engineering.

## 11. Implementation Technology

Just to recall, the project consists of a web application, server and a database. We should choose some reliable technology to implement each one. We wanted to find solutions that are efficient, cheap but also that are familiar to our colleagues.

Front-end can be implemented with the help of React. This choice has been made when we took into consideration that about half of the students had just taken a course on React, so we could assign web application development to these groups, other than that it's needless to say that React is a popular frontend development tool, which means we could receive support from the community in case of any troubles and it also means that many problems already have some solution that can be found on stackoverflow for example. Moreover there are many free to use components that will save us some work. React also offers great performance, easy testing solutions and is SEO friendly, which increases the chances to appear on the first page in google search results.

For the most part, we have been learning C-based languages, that's why we would like to use Asp.NET so that those of us who prefer to code in C could take care of the back-end. There are many other great options like Django or Node.js, however Asp.NET is a great choice, it is popular, which makes finding solutions much easier, it supports JIT (a method of executing code with compilation during the execution) compilation, that reduces memory usage. Additionally, this backend technology offers built-in configuration information, which makes Asp.NET easy to deploy, it also is reliable, fast and offers a vast database support.

MySQL is our choice for a database. The benefits are that MySQL is free and open-source but it also offers high performance and high reliability, it ties in well with ASP.NET and again some of us are already familiar with it.

We are open to different propositions, depending on the preferences of our colleagues, choosing familiar technologies comes with a lot of benefits but it's not unusual to be eager to

learn a new technology.

## **12. Final Words**

The scope of this project is rather too large for a single semester, however it can be easily scaled down, the main functionality is the process of going through session, not all methods must be implemented right away, 5 would be a good start, ideally it would be great if the methods were implemented in an interactive, creative way to encourage users to use the application, however that's not necessary. Statistics are nice to have, but they can be implemented at other point in time.

## **13. Glossary**

- TEAM-CBT - a framework for conducting measurement-based therapy that is proven to be more effective than traditional therapy.
- User - Person using the web application, whether it's someone looking to get help or an admin.
- Session - Process of going through the Testing, Empathy, Agenda Setting, Methods steps of TEAM-CBT
- Method - a technique used to reduce user's belief in a negative thought.
- Evaluation - a mood test that the user can take in one of the following categories: Depression, Anxiety, Relationships, Addictions, Happiness in the form of simple questions that are scored from 0 to 4.

## 14. Appendix A

Fig. 37. Methods depending on found Distortions

## Checklist of Negative and Positive Distortions\*

Distortion	Negative Distortion Example	Positive Distortion Example
<b>1. All-or-Nothing Thinking.</b> You think about yourself or the world in black-or-white, all-or-nothing categories. Shades of gray do not exist.	When you fail, you may tell yourself that you're a complete failure.	When you succeed, you may tell yourself that you're a winner and feel superior.
<b>2. Overgeneralization.</b> You think about a negative event as a never-ending pattern of defeat or a positive event as a never-ending pattern of success.	When you're rejected by someone you care about, you may tell yourself that you're an unlovable loser who will be alone and miserable forever.	When you overcome an episode of depression or self-doubt, and you're suddenly feeling happy again, you may tell yourself that all your problems are solved and that you'll always feel happy.
<b>3. Mental Filter.</b> You think exclusively about your shortcomings and ignore your positive qualities and accomplishments. Or, you dwell on the positives and overlook the negatives.	A TV talk show host told me that he typically received hundreds of enthusiastic emails from fans every day, but there was nearly always one critical email from a disgruntled viewer. He explained that he'd obsess for hours about the negative email and completely overlook the hundreds of glowing ones. As a result, he constantly struggled with feelings of anxiety and low self-esteem in spite of his tremendous ratings and popularity.	You may fantasize about how good that dessert will taste, and ignore the negatives, like gaining weight and feeling guilty or bloated afterwards. Or, you may tell yourself how great you'll feel if you have a drink, and ignore the fact you nearly always drink too much and end up with a hangover.
<b>4. Discounting the Facts.</b> You tell yourself that negative or positive facts don't count, so as to maintain a universally negative or positive self-image.	<b>Discounting the Positive:</b> When someone genuinely compliments you, you may tell yourself they're only saying that to make you feel good.	<b>Discounting the Negative:</b> When you're trying to diet and feeling tempted by something tasty, you may tell yourself, "I'll only have one little bite." But you've probably given yourself this message on hundreds of occasions, and it has never once been accurate! During an argument, you may get defensive and insist that the other person is "wrong." Then the conflict escalates.
<b>5. Jumping to Conclusions.</b> You jump to conclusions that aren't warranted by the facts. There are two common forms:  <b>Mind-Reading</b> , you make assumptions about how other people are thinking and feeling. <b>Fortune-Telling</b> , you make dogmatic negative or positive predictions about the future.	<b>Mind-Reading:</b> If you're feeling shy at a party, you may tell yourself that other people don't have to struggle with shyness or that they'd look down on you if they knew you were shy.  <b>Fortune-Telling:</b> When you're depressed, you may tell yourself that you'll never recover. When you're feeling anxious, you may tell yourself that something <i>terrible</i> is about to happen—"When I give my talk, my mind will go blank. I'll look like an idiot."	<b>Mind-Reading:</b> You may tell yourself that a relationship is going really well when the other person is actually feeling annoyed or unhappy with you.  <b>Fortune-Telling:</b> You may tell yourself, "I'll just have one drink" or "one bite," when, in fact, you <i>never</i> stop at just one drink or bite.

\* Copyright © 2010 by David D. Burns, M.D. Do not reproduce without permission. Revised 2014.

Fig. 38. List of Cognitive Distortions

<b>6. Magnification and Minimization.</b> You blow things out of proportion or shrink their importance inappropriately. This is also called the "binocular trick" because it's like looking through the ends of a pair of binoculars, so things either look much bigger, or much smaller, than they are in reality.	When you're procrastinating, you may think about <i>everything</i> that you've been putting off and tell yourself how <i>overwhelming</i> all those tasks will be. (Magnification) You may also tell yourself that you're efforts today wouldn't amount to anything anyway, so you might as well put it off. (Minimization)	When you're trying to diet and you're feeling tempted, you may tell yourself: "This ice cream will taste so good!" (Magnification). Will it <i>really</i> be that good? Will it be worth the way you'll feel about yourself after you give in to the urge to binge?
<b>7. Emotional Reasoning.</b> You reason from how you feel. In point of fact, your feelings result from your thoughts, and not from what's actually happening. If your thoughts are distorted, your feelings will be as misleading as the grotesque images you see in curved funhouse mirrors.	You may tell yourself, "I'll clean my desk (or start my diet) when I'm more in the mood. I just don't <i>feel</i> like it right now." Or course, the feeling never comes! When you're depressed, you may tell yourself, "I <i>feel</i> like a loser, so I must <i>really</i> be one." Or "I <i>feel</i> hopeless, so I must be hopeless."	When you're gambling, you may say, "I <i>feel</i> lucky! I just know I'm about to hit the jackpot." This distortion also triggers romantic intoxication. When you meet someone attractive, you may feel so happy and excited that you think that he or she <i>must be</i> wonderful—the man (or woman) of your dreams.
<b>8. Should Statements.</b> You make yourself (or others) miserable with "shoulds," "musts" or "ought to's."	<b>Self-Directed Shoulds:</b> You tell yourself that you <i>shouldn't</i> have screwed up and made such a stupid mistake. <b>Other-Directed Shoulds:</b> You may tell yourself, "That fellow <i>shouldn't</i> cut in front of me in traffic like that. I'll show him that he can't get away with it!" <b>World-Directed Shoulds:</b> "The train <i>shouldn't</i> be late when I'm in such a hurry!"	<b>Self-Directed Shoulds:</b> When you're feeling tempted, you may tell yourself, "I've had a hard day. I deserve a drink (or a nice dish of ice cream)." <b>Other-Directed Shoulds:</b> You may tell yourself that your values are the best values and that other people should think and feel the same way. <b>World-Directed Shoulds:</b> You may tell yourself that the world should be the way you expect it to be.
<b>9. Labeling.</b> You label yourself or others instead. Labeling is actually an extreme form of overgeneralization, because you see your entire self or essence as defective and globally bad, or superior.	You may label yourself or someone you're not getting along with as "a loser" or "a jerk." A physician slipped up on her diet and gave in to the temptation to eat a donut. Then she told herself that she was "a fat pig with no will power." This thought was so upsetting that she ate six more donuts.	When you do well, you may think of yourself as special or as "a winner." Motivational speakers, politicians, and athletic coaches often use this strategy to motivate people. But in reality, there's no such thing as a "winner" or a "loser." We're all human beings, and no one can win or lose all the time.
<b>10. Blame.</b> You find fault with yourself ( <b>Self-Blame</b> ) or others ( <b>Other Blame</b> ).	<b>Self-blame.</b> If you're depressed, you may beat up on yourself constantly and mercilessly, blaming yourself for every error and shortcoming instead of using your energy to find creative solutions to your problems.	<b>Other-blame.</b> During an argument, you may tell yourself that the other person is to blame for the conflict. Then you feel like an innocent victim and overlook your own role in the problem.

\* Copyright © 2010 by David D. Burns, M.D. Do not reproduce without permission. Revised 2014.

Fig. 39. List of Cognitive Distortions

## 15. Appendix B

**Instructions:** Use checks (✓) to indicate how you're feeling *right now*.

**Answer all items.**

Depression	0—Not at all	1—Somewhat	2—Moderately	3—A lot	4—Extremely
1. Sad or down in the dumps					
2. Discouraged or hopeless					
3. Low in self-esteem, inferior, or worthless					
4. Unmotivated to do things					
5. Decreased pleasure or satisfaction in life					
<b>Total Items 1 to 5 ➔</b>					

Fig. 40. Depression Test

Anxiety					
1. Anxious					
2. Frightened					
3. Worrying about things					
4. Tense or on edge					
5. Nervous					
<b>Total Items 1 to 5 →</b>					

Fig. 41. Anxiety Test

Score	Severity	Meaning
0	No symptoms	That's terrific! You don't seem to have any symptoms at all.
1–2	Borderline	These scores are normal, but you could use a little tune-up.
3–5	Mild	Although your scores are not greatly elevated, this is enough depression or anxiety to take the joy out of life. If we work together, we can probably get your scores down to 0, which would be terrific!
6–10	Moderate	You're feeling quite a bit of depression or anxiety. Although you're not in the severe range, these scores reflect considerable unhappiness.
11–15	Severe	You have fairly strong feelings of depression or anxiety. That makes me sad, but there's some really good news. The tools in this book can help you transform your negative feelings into joy.
16–20	Extreme	Scores in this range indicate that your suffering is intense. Friends or family may have trouble grasping how much pain you're in. The good news is that the prognosis for improvement is very positive. In fact, recovery is one of the greatest feelings a human being can have.

Table 23. Meaning of score of depression and anxiety tests

**Instructions:** Use checks (✓) to indicate how you're feeling *right now*.

**Answer all items.**

Anger	0—Not at all	1—Somewhat	2—Moderately	3—A lot	4—Extremely
1. Frustrated					
2. Annoyed					
3. Resentful					
4. Angry					
5. Irritated					
<b>Total Items 1 to 5 ➔</b>					

Fig. 42. Anger Test

Think about an important relationship, like your spouse, partner, friend, colleague, or family member. Use checks (✓) to indicate how you feel about this relationship.

**Answer all items.**

0—Very dissatisfied	1—Moderately dissatisfied	2—Somewhat dissatisfied	3—Neutral	4—Somewhat satisfied	5—Moderately satisfied	6—Very satisfied
1. Communication and openness						
2. Resolving conflicts						
3. Degree of affection and caring						
4. Intimacy and closeness						
5. Overall satisfaction						
<b>Total Items 1 to 5 ➔</b>						

Fig. 43. Relationships Test

Anger Scale		Relationship Satisfaction Scale	
Score	Meaning	Score	Meaning
0	No anger	0–10	Extremely dissatisfied
1–2	A little anger	11–15	Very dissatisfied
3–5	Mild anger	16–20	Moderately dissatisfied
6–10	Moderate anger	21–25	Somewhat satisfied
11–15	Severe anger	26–28	Moderately satisfied
16–20	Extreme anger	29–30	Extremely satisfied

Table 24. Meaning of score of relationship and anger tests

!

**Instructions:** Use checks (✓) to indicate how much each statement describes how you have been feeling in the past week, including today.

**Answer all items.**

	0—Not at all	1—Slightly true	2—Moderately true	3—Very true	4—Completely true
1. Sometimes I crave drugs or alcohol.					
2. Sometimes I have the urge to use drugs or alcohol.					
3. Sometimes I really want to use drugs or alcohol.					
4. Sometimes it's hard to resist the urge to use drugs or alcohol.					
5. Sometimes I struggle with the temptation to use drugs or alcohol.					
<b>Total Items 1 to 5 →</b>					

Fig. 44. Addictions Test

**Instructions:** Use checks (✓) to indicate how you're feeling right now.

**Answer all items.**

	0—Not at all	1—Somewhat	2—Moderately	3—A lot	4—Extremely
1. Happy and joyful					
2. Hopeful and optimistic					
3. Worthwhile, high self-esteem					
4. Motivated, productive					
5. Pleased and satisfied with life					

**Total Items 1 to 5 ➔**

Fig. 45. Happiness Test

Score	Happiness Level	Meaning
0–1	No happiness	It seems like you're barely having any positive feelings at all right now. That's really sad, but there's some good news—if you want, we can work together and fix that.
2–4	Minimal happiness	These scores indicate that you have very few positive feelings. There's lots of room for improvement.
3–5	Some happiness	Your feelings are somewhat positive, which is promising. If we work together, we should be able to make things a lot better.
6–10	Moderate happiness	You seem to be feeling moderately positive. That's good! I'd love to see your scores increase even more.
11–15	A lot of happiness	You seem to be feeling very positive and happy, but there's room for feeling even happier.
16–19	Extreme happiness	Scores in this range are really good. You're feeling extremely positive in at least one of the five areas on the test. Way to go! There's still a little room for feeling even better.
20	Tremendous happiness	This is fabulous!

Table 25. Meaning of score of happiness test

## 16. Further Reading

- A Brain User's Guide to TEAM Therapy\* - explanation of why TEAM-CBT works by Mark Noble - a pioneering researcher in the fields of stem cell biology and stem cell medicine.<sup>†</sup>
- Exploring the Daily Mood Log<sup>‡</sup> - overview of why the Daily Mood Log is "one of the greatest inventions in the history of psychology" by Mark Noble.
- Feeling Great - David Burns' book on TEAM-CBT.<sup>‡</sup> The description of each method can be found at the end of the book, these are the methods that the project is going to use.
- The Anxiety Encyclopedia: Your Path to Recovery by Jotham Sadan - why anxiety happens and how to get out of it with science (it's a different approach, not TEAM-

---

\* <https://feelinggood.com/wp-content/uploads/2021/12/AAAB-Brain-Users-Guide-to-TEAM-Therapy.pdf>

† <https://www.urmc.rochester.edu/people/23095977-mark-david-noble>

‡ <https://feelinggood.com/wp-content/uploads/2021/12/AAA-Exploring-the-Daily-Mood-Log.pdf>

‡ <https://www.amazon.com/Feeling-Great-Revolutionary-Treatment-Depression/dp/168373288X>

CBT but MBCT, has got a great chapter on science of why we get anxious)<sup>§</sup>

- Feeling Good Together: The Secret to Making Troubled Relationships Work by David Burns<sup>¶</sup>
- Feeling Good Podcasts<sup>††</sup>

## List of Tables

1	User Stories for Testing . . . . .	12
2	User Stories for Periodic Tests . . . . .	12
3	User Stories for Methods . . . . .	13
4	User Stories for Methods . . . . .	14
5	User Stories for Agenda Setting . . . . .	14
6	User Stories - General . . . . .	15
7	User Stories for Admin . . . . .	15
8	Reverse Side for "fill out the recovery circle" . . . . .	16
9	Reverse Side for "be able to look at quick summary of chosen technique" . . . . .	16
10	Reverse Side for "receive suggestions regarding what methods to choose" . . . . .	17
11	Reverse Side for "be able to use the ... Technique" . . . . .	17
12	Reverse Side for "write down the specific event that caused my upset" . . . . .	17
13	Reverse Side for "check what feelings from: ... am I feeling" . . . . .	18
14	Reverse Side for "write down my negative thought" . . . . .	18
15	Reverse Side for "have an easy way to look up a cheat sheet with cognitive distortions" . . . . .	18
16	Reverse Side for "fill out the depression/anxiety/relationships/happiness/addictions checklist" . . . . .	19
17	Reverse Side for "manage my account" . . . . .	19
18	Reverse Side for "be able to set up an account" . . . . .	19
19	Reverse Side for "be able to login" . . . . .	20
20	Reverse Side for "see the statistics of taking a particular test" . . . . .	20
21	Reverse Side for "have access to see the total data in a chart" . . . . .	20
22	Reverse Side for "reject some tests" . . . . .	21
23	Meaning of score of depression and anxiety tests . . . . .	82
24	Meaning of score of relationship and anger tests . . . . .	84
25	Meaning of score of happiness test . . . . .	86

## List of Figures

1	Daily Mood Journal . . . . .	5
2	Daily Mood Journal . . . . .	6
3	Recovery Circle . . . . .	7
4	Positive Reframing . . . . .	8
5	Use Case Diagram for General module . . . . .	22

<sup>§</sup> <https://www.amazon.com/Anxiety-Encyclopedia-Your-Path-Recovery/dp/B08RX78PGR/>

<sup>¶</sup> <https://www.amazon.com/Feeling-Good-Together-Troubled-Relationships/dp/0767920821/>

<sup>††</sup> <https://feelinggood.com/category/dr-davids-blogs/feeling-good-podcast/>

6	Use Case Diagram for Testing module . . . . .	23
7	Use Case Diagram for Agenda module . . . . .	23
8	Use Case Diagram for Methods module . . . . .	24
9	Use Case Diagram for Admin module . . . . .	25
10	Class Diagram . . . . .	29
11	State Diagram for User . . . . .	33
12	State Diagram for Session . . . . .	33
13	State Diagram for Server . . . . .	34
14	State Diagram for Evaluation . . . . .	35
15	State Diagram for frontend . . . . .	36
16	Activity Diagram for Login Use Case . . . . .	38
17	Activity Diagram for New Session Use Case . . . . .	39
18	Activity Diagram for See Stats Use Case . . . . .	40
19	Activity Diagram for Registration Use Case . . . . .	41
20	Activity Diagram for Evaluation Use Case . . . . .	42
21	Activity Diagram for Continue Previous Session Use Case . . . . .	42
22	Activity Diagram for Manage Accounts Use Case . . . . .	43
23	Activity Diagram for Manage Account Use Case . . . . .	43
24	Activity Diagram for Show Test Results Use Case . . . . .	44
25	Sequence Diagram for New Session Use Case . . . . .	47
26	Sequence Diagram for Continue Previous Session Use Case . . . . .	48
27	Sequence Diagram for Login Diagram . . . . .	49
28	Sequence Diagram for Register Diagram . . . . .	50
29	Sequence Diagram for See Statistics Use Case . . . . .	51
30	Sequence Diagram for Start Evaluation Use Case . . . . .	52
31	Sequence Diagram for Manage Account Use Case . . . . .	52
32	Sequence Diagram for User Retrieval . . . . .	53
33	Sequence Diagram for Show Test Results Use Case . . . . .	53
34	Sequence Diagram for displaying Main Page . . . . .	54
35	Sequence Diagram for Filtering . . . . .	54
36	Sequence Diagram for Banning Users . . . . .	55
37	Methods depending on found Distortions . . . . .	79
38	List of Cognitive Distortions . . . . .	80
39	List of Cognitive Distortions . . . . .	81
40	Depression Test . . . . .	81
41	Anxiety Test . . . . .	82
42	Anger Test . . . . .	83
43	Relationships Test . . . . .	83
44	Addictions Test . . . . .	84
45	Happiness Test . . . . .	85