BABEȘ-BOLYAI UNIVERSITY CLUJ-NAPOCA

FACULTY OF MATHEMATICS AND COMPUTER SCIENCE

SPECIALIZATION MATHEMATICS AND COMPUTER SCIENCE IN ENGLISH

DIPLOMA THESIS

Mental Health Screening Tool for Students
with Intelligent Data Analysis

Supervisor

Lecturer, PhD. Adriana M. COROIU

Author

Amalia-Maria POSTOLACHE

2023

UNIVERSITATEA BABEȘ-BOLYAI CLUJ-NAPOCA

FACULTATEA DE MATEMATICĂ ȘI INFORMATICĂ

SPECIALIZAREA MATEMATICĂ-INFORMATICĂ ÎN LIMBA ENGLEZĂ

LUCRARE DE LICENȚĂ

Instrument de Examinare a Sănătății Mentale a Studenților cu Analiză Inteligentă a Datelor

Conducător științific:

Lector Universitar, Adriana M. COROIU

Absolvent

Amalia-Maria POSTOLACHE

2023

Abstract

This paper aims to understand the nexus between academic context and student well-being. It provides an overview of the prevalence of mental health problems among students and the ways in which these issues can impact academic performance. Nevertheless, the factors of the academic environment that can influence a student's mental health are also discussed, outlining the importance of addressing such issues in the student community and the potential long-term impact it can have on individuals, leading us to the reasoning behind creating the application that serves as the focus of this paper. After doing extended research to understand the role of mental health in the academic context and using the domain-related knowledge I have been equipped with during my years of study, I was able to develop an application that has as its focus a mental health quiz designed for students and additional functionalities and resources meant to provide adequate assistance in helping students cope with the challenges met in the process of academic functioning. Considering the practical use of the application and what it offers, this paper will also discuss the intelligent analysis of the collected data, enabling further interpretation by mental health professionals.

Contents

List of Figures

Chapter 1

Introduction

1.1 Motivation

Living in a world where the professional success of a high school graduate can be predicted by his choice to pursue higher education studies and the career prospect is directly impacted by his academic performance, I was determined to find a way of supporting the students who do not always meet the expectations of either their families or their educators. A student who drops out of school embodies the image of incompetence or disinterest most of the time, but how often have people taken the time to analyze the factors that led to the dropout or the poor academic performance that often precedes it?

A student's academic functioning and, subsequently, career outlook are unfortunately heavily influenced by mental health disorders, given that most factors weighing in on a student's mental health state originate in the academic environment. These factors embody academic expectations and pressure to succeed, meet deadlines, manage time efficiently, or even cover the cost of education.

1.2 Objective

I decided to focus on what can be done to ensure a stress-free, career-oriented academic experience for all types of students, regardless of intellectual, emotional, or financial background, while also giving students the opportunity to learn more about their struggles, how to manage their emotions and difficulties, and seek support from a specialist.

This study focuses on the process of building an application that is user(or student)-friendly, which revolves around the mental health quiz created in collaboration with a certified psychologist. In addition to the self-assessment process of the quiz, the student is able to monitor its progress and receive an objective perspective on its mental health state, either by seeking support from a specialist or by learning more from the resources available within the application. In order to address also a long-term solution, the student is able to subscribe to a newsletter meant to provide assistance in self-care and time or stress management.
Additionally, the data obtained from quiz completion by users is subject to intelligent analysis to ensure use by a mental health professional in case of further development.

1.3 Structure of the thesis

This paper is divided into three main chapters: Theoretical Background, Mental Health Checker for Students, and Conclusions.

Firstly, we introduce the theoretical background for the field behind the concept of the application, briefly discussing the general context of the study of mental health status in college students. An input on how the COVID-19 pandemic has affected and influenced the academic experience and performance is another topic from this chapter, before moving on to the research that has been done and the measures taken following the research. After gaining an insight into the approach to the study of students' mental health in the last few years, we discuss some existing solutions and how they have made a change in the topic of mental health.

Secondly, we focus on the application that is the object of the thesis, starting with the concept of the application and its objectives. We will also be discussing details of the implementation, the technologies that have been used in the creation process, the available functionalities within the application, and the overall user experience. Furthermore, the intelligent data analysis topic and the training and testing processes will also be presented.

Lastly, a summary of the paper's content, while highlighting the final conclusions, will be presented. While presenting the progress made throughout the development process of the thesis, the improvements that can be made to the application in the context of further development will also be discussed.

Chapter 2

Theoretical Background

2.1 General context surrounding students' mental health

Mental health is an important aspect of overall well-being, and it can be particularly challenging for students to maintain good mental health due to the many stresses and demands of the academic environment. In recent years, college student populations have witnessed a surge in symptoms of depression, anxiety, and other mental illnesses, accompanied by a consistent growth in the demand for counseling services [1]. College students frequently experience mental health issues, which are clearly connected with lower academic performance. Additional research is required to determine whether this link could be causal and, if so, whether therapies aimed at addressing mental health problems could enhance academic performance.

The college years are a critical developmental stage in which students transition from late adolescence to emerging adulthood [2]. Some of the long-term adverse outcomes (including persistent emotional and physical health problems or relationship dysfunction) may be influenced by mental health issues that arise throughout the college years, as these years are a peak phase for the emergence of a wide spectrum of mental disorders.

According to research conducted by the Romanian National Council for Higher Education Financing in 2018, around 74% of Romanian high school graduates pursued their studies in higher education. A study by the Romanian Ministry of Education revealed that, in the 2019-2020 academic year, the overall dropout rate for Romanian universities was 10.4%, which was a slight decrease from the previous year's rate of 10.5%. Previous research has found that

college students with mental disorders or illnesses are twice as likely to drop out without obtaining a degree.

There are fewer studies that focus on the association between mental health problems and academic success in college. The majority of data supports the conclusion that depression and suicidal thoughts and behaviors are associated with a lower grade point average [3].

2.2 COVID-19 Impact on Mental Health in College Students

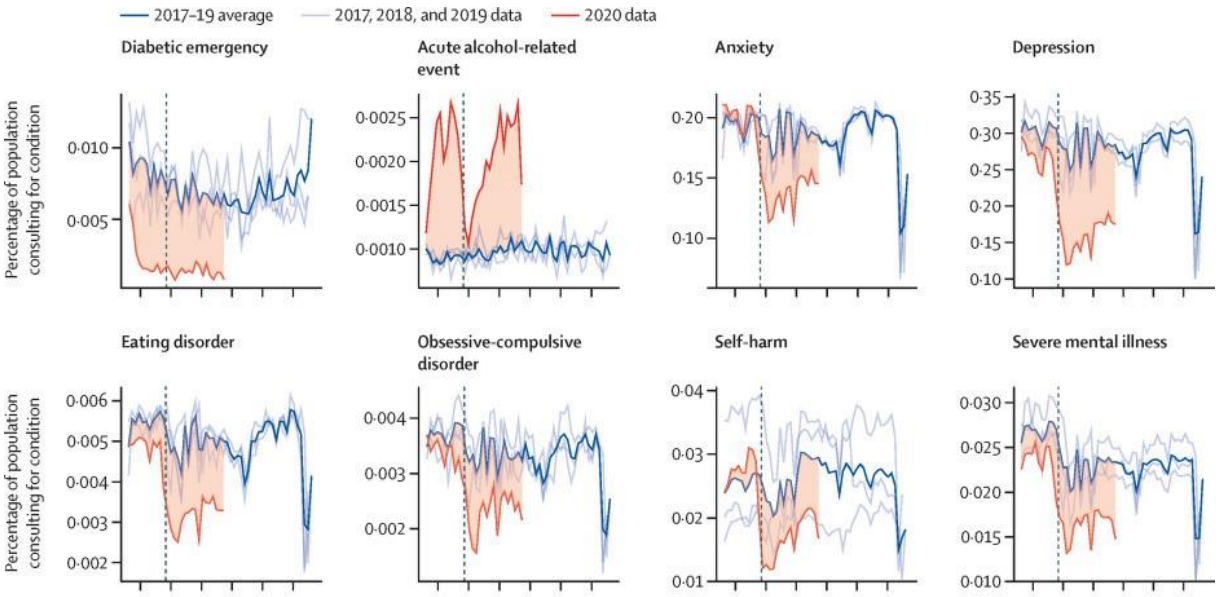

Figure 2.2.1: Proportions of each study population with contacts for mental health conditions in 2017–19 and 2020 [4]

The COVID-19 pandemic has spread quickly in most of the world's nations and has had unanticipated effects on health, the economy, society, education, and psychology. Emergencies can have an impact on the public's well-being, safety, and health (by causing confusion, insecurity, stigma, and emotional isolation). Furthermore, communities may experience school closures, job losses, food insecurity, inadequate distribution of essential items, and insufficient resources for medical response. The closure of schools and resulting educational disruptions during the pandemic have had a significant impact on the psychological well-being of students (as can be seen in Figure 2.2.2 by analyzing the 11-30 age group). According to The European

University Association (EUA), it has been reported that 90% of European universities experienced delays in student progression and completion due to the pandemic, with 60% reporting disruptions in student learning and 45% reporting negative impacts on the mental health of students [5].

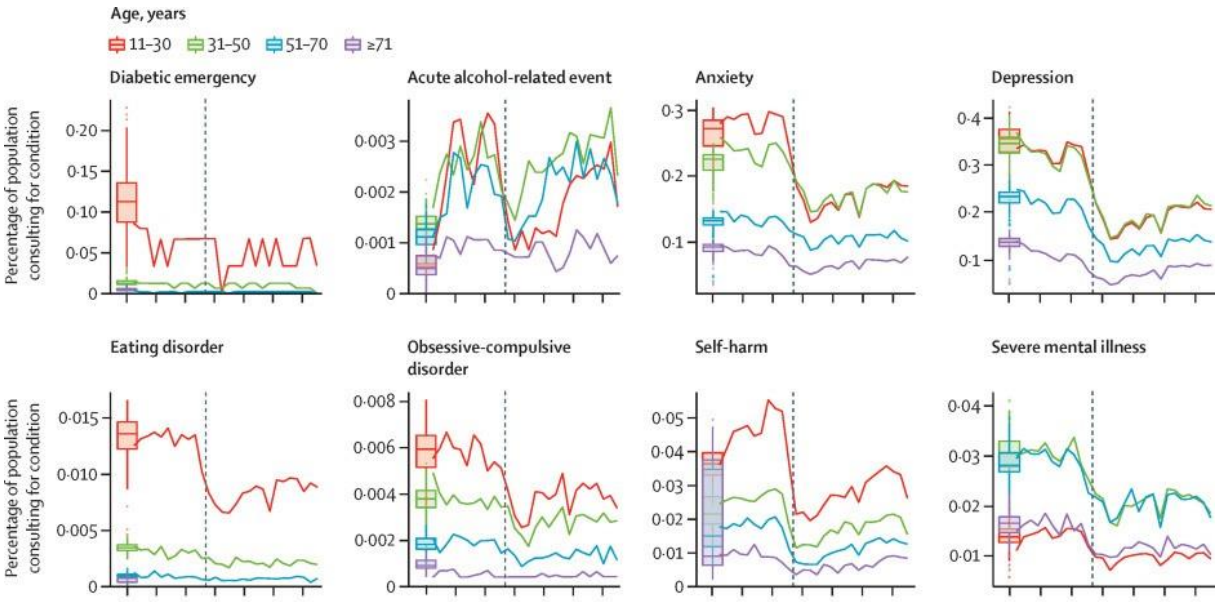

Figure 2.2.2: Percentage of each denominator population with general practitioner contacts for the study conditions throughout 2020, by age group [4]

The transition to online learning and the lack of access to physical resources have contributed to a decline in academic performance among Romanian students as well. Some students have reported a lack of motivation and difficulty concentrating, while others have experienced technical difficulties or limited access to academic resources. Additionally, the uncertainty surrounding the pandemic, financial difficulties, and difficulties in accessing mental health support have further compounded the situation, highlighting the need for adequate support and resources to help students cope with these challenges.

A survey conducted by the Romanian National Alliance of Student Organizations found that 71% of Romanian students experienced an increase in stress levels due to the pandemic, while 61% reported anxiety and 38% reported depression. Another survey conducted by the Babes-Bolyai University in Cluj-Napoca revealed that 84% of respondents reported a negative impact on their mental health due to the pandemic, with 77% reporting a decline in academic performance, and a study conducted by the University of Bucharest found that Romanian

students experienced high levels of stress and anxiety during the pandemic, with those in the medical and social sciences reporting the highest levels of stress. Some of the results of this study can be seen in Figure 2.2.3.

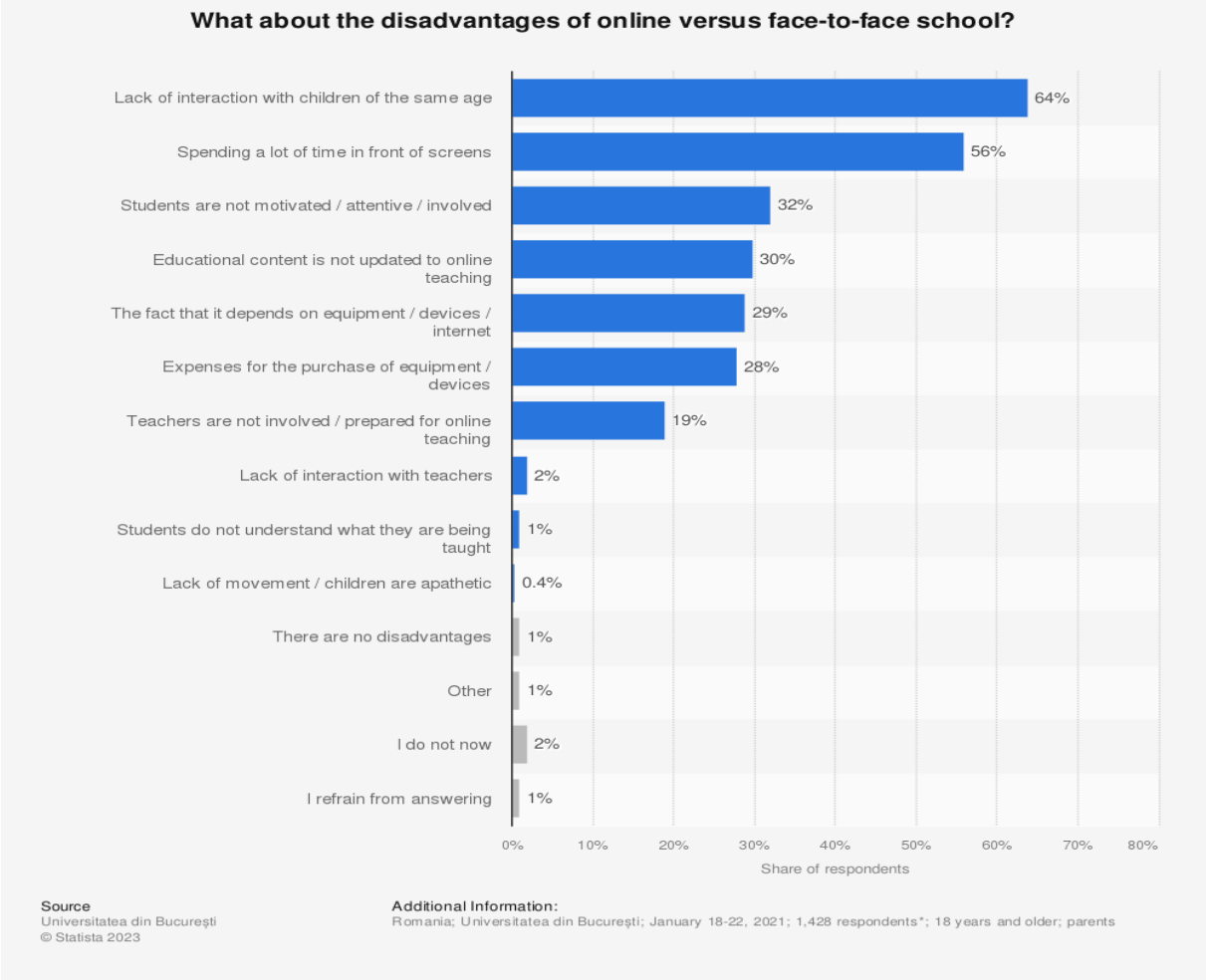

Figure 2.2.3: Starting school in a pandemic context survey [6]

2.3 Research and measures taken

A study conducted by the researchers of the Franciscan University in Steubenville, Ohio, has indicated that the top three concerns of students between the ages of 18 and 24 were academic performance, the drive and pressure to succeed, and post-graduation plans and aspirations. According to a 2019 study by the National Alliance on Mental Illness, 75% of all lifetime cases of mental illness begin by age 24. While stress is an unavoidable aspect of life, it is very present and becoming more prevalent among university students [7]. In addition to anxiety and stress, depression may have a significant influence on college life, necessitating extensive study to assist future students. In Europe, about 10% to 25% of the student population has been diagnosed with or treated for depression. As obtaining a college degree is viewed as the key to success, more students leave their hometowns to pursue higher education. This change, however, has the potential to cause emotions of melancholy, worry, and tension. This shift has been associated with concentration problems or social isolation; therefore, homesickness is a direct byproduct that could exacerbate mental health issues in university students and should be a significant emphasis for universities attempting to solve the difficulties impacting their students.

Academic pursuits are a fundamental part of a student's life, and having a negative attitude towards the goals of college life can lead to overwhelming levels of stress. The pressure of meeting academic expectations, such as achieving specific grades, meeting deadlines, taking tests, managing a large volume of course material, and organizing time effectively, has been demonstrated to be a major cause of stress and can lead to feelings of overwhelm, burnout, and depression. Additionally, the cost of education can be a significant burden, particularly for students who are supporting themselves or their families. The high cost of tuition fees, textbooks, and other educational expenses can also impact a student's ability to balance work and academics, given that, according to a Eurostat report from 2021, around 45% of students in Europe were employed while studying to cover the cost of their education [8]. The career prospect may also be affected by this factor, given that students who cannot afford to pay for their education may miss out on various opportunities such as internships or study abroad programs or reach a stage where the financial requirements to continue their studies cannot be met. While the academic experience can be perceived as a positive challenge with chances of

increasing competency and learning capacity, if perceived negatively, this stress can be detrimental to the student's mental health [9].

To customize treatments to their students' particular requirements, universities must first investigate what other aspects of life, other than academics, may be causing this increase in mental health issues. For example, low self-esteem, low satisfaction in life, and feelings of inferiority can significantly affect social and educational functioning [10]. Students who grew up in financially unstable households are more likely to experience symptoms of depression and anxiety, indicating that financial pressures are correlated with higher rates of mental health issues in students. According to the researchers, poor sleeping habits have a detrimental influence on academic progress as well as mental health concerns.

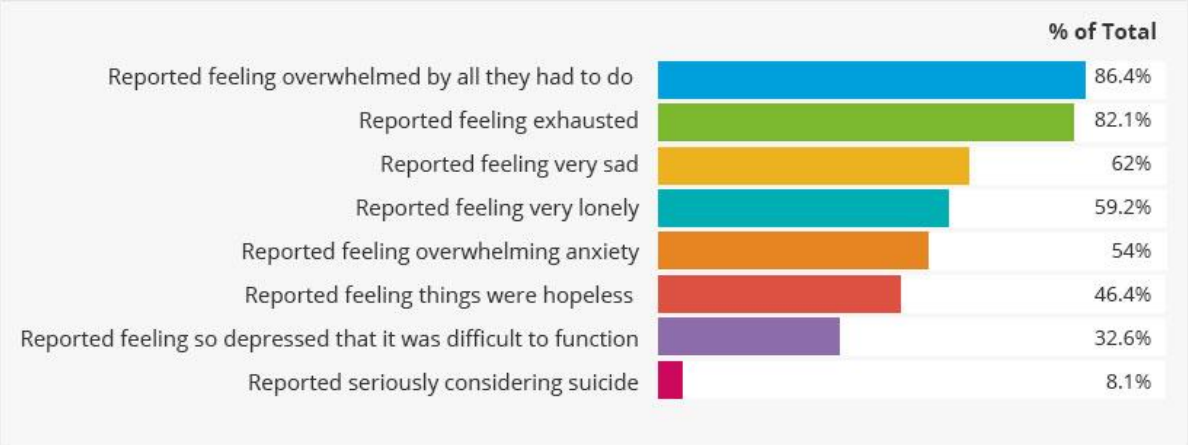

Figure 2.3.1: What goes through a typical college student's mind over a twelve-month period [11]

It is crucial to address the occurrence of depression, anxiety, and stress among college students due to the negative side effects and consequences they can have. Depression is linked to detrimental behaviors such as fatigue or loss of energy, changes in appetite or sleep patterns, a lack of exercise, and failure to follow medical treatment guidelines. People with anxiety disorders also report a lower quality of life than those who do not have significant levels of anxiety [12]. Anxiety can interfere with a student's ability to participate in social activities and form relationships, which can contribute to feelings of loneliness and isolation, but it can also lead to difficulties with concentration, memory, and decision-making, impacting the career prospect. Gaining insight into the aspects of life that correspond with a decrease in symptoms of depression, anxiety, and stress can be valuable for educational institutions, so those behaviors can be encouraged in their students. For instance, studies have reported that satisfactory

connections with family and friends are more likely to generate overall life satisfaction among college students [13].

According to a 2019 report from the European Students' Union, there has been a significant increase in the demand for mental health services among students in Europe in recent years. The report notes that "students in Europe increasingly suffer from mental health issues such as anxiety, depression, and burnout" and that "universities and student support services have reported a surge in demand for counselling and mental health services." It is important to address the mental health stigma in students given the potential long-term impact it can have on individuals and on society.

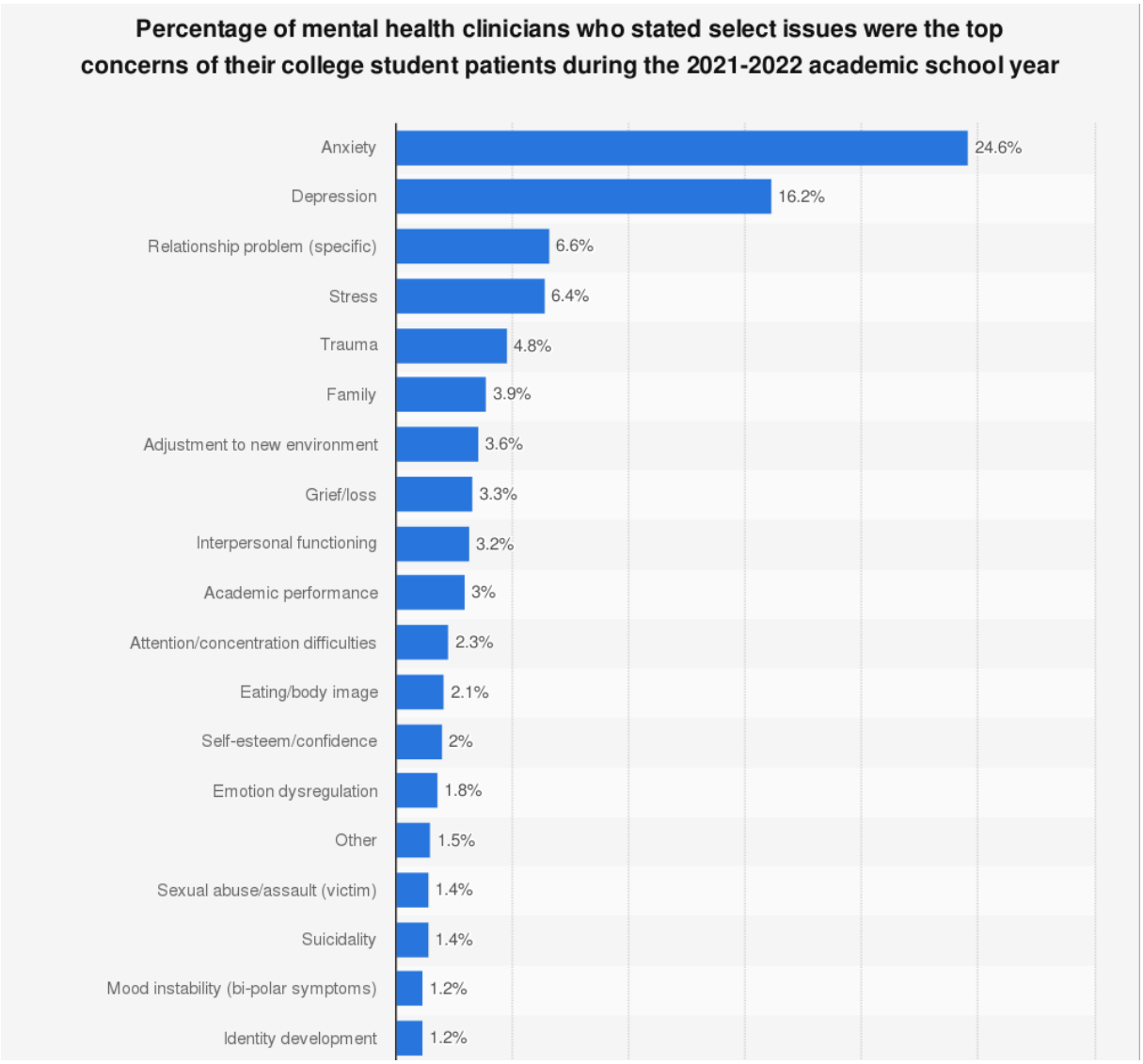

Figure 2.3.2: Percentage of mental health clinicians who stated select issues were the top concerns of their college student patients during the 2021-2022 academic school year [14]

Despite increased focus on student mental health, the impact of counselling is still understudied at European level, given the failure to prioritise mental well-being in general.

Education ministers from across the European Higher Education Area (EHEA) supported the "*Principles and Guidelines to Strengthen the Social Dimension of Higher Education in the EHEA*" during the Rome 2020 Ministerial Conference, thereby pledging to develop policies that would allow higher education institutions to provide helpful advice and counselling for students. According to the Eurydice report, "*Towards equity and inclusion in higher education in Europe*", psychological counselling services are legally required in only about half of the countries in the European Higher Education Area. The "*Well-being"* program provided by the Ghent University of Belgium, "*Disabilities and psychological counselling"* by the University of Warsaw, Poland, and "*Student well-being"* by the Vrije Universiteit Amsterdam of The Netherlands are among the counselling services offered to students in some European countries that are also part of the European Education Area. Even if many countries have made counselling services and professional interventions widely available, the essential point is often neglected: many students who suffer from clinically significant mental disorders and suicidal thoughts do not receive treatment because they do not believe that they need it. More so, some other obstacles that prevent students from seeking mental health support include the pressure of a heavy study workload and the dread of being stigmatised. In this context, it is important for national authorities and universities to reevaluate their existing methods and explore new ways of reaching out to vulnerable or disadvantaged students.

Given the context, the Flemish Community (Belgium) has implemented a distinct and durable strategy for the mental health of students that prioritizes the public mental health perspective. The strategy is based on scientific epidemiological data that identifies the needs of the full student body and focuses on addressing both the incidence and prevention of mental health issues. This strategy changes the emphasis away from mental illnesses and high-risk students and toward the emotional well-being of the whole student population, and as a result, incorporates treatments that can improve the resilience of all students. Improving the students' psychological resilience will also reduce the incidence of students developing mental disorders when they encounter sources of stress. Improved resilience skills can serve as a safety net for students when they face extraordinary risk factors, such as the COVID-19 pandemic.

A collaborative project called "Peer Learning Activities and Resources for Advancing Social Dimension in the European Higher Education Area (EHEA)" is underway. In partnership with the Bologna Working Group on the Social Dimension, the project aims to facilitate peer-learning events that foster the exchange of best practices among EHEA countries at various stages of implementation. The goal is to enhance and sustain the knowledge sharing process, ultimately advancing the principles and guidelines for the social dimension across the EHEA. Considering the identified requirements and adopting a public mental health standpoint, this project aims to create a complete framework for action that comprises a self-evaluation instrument and recommendations to assist in establishing long-lasting strategies for enhancing student mental health in higher education at both national and institutional levels, as well as a central scholarship web portal intended to serve as a financial assistance counselling and informational tool for students [15].

2.4 State of the art

Applications dedicated to the mental health of students (and not only) have gained significant attention and popularity in recent years. These applications aim to support students' well-being by providing accessible tools, resources, and assessments related to mental health. Some of the existing approaches in this domain cover the following functionalities:

- Self-assessment and monitoring
  Many mental health applications for students offer self-assessment tools and daily mood tracking features. These tools help students gain insight into their emotions, stress levels, and overall well-being.

- Guided practices and exercises
  Some applications provide guided practices and exercises to promote relaxation, mindfulness, and stress reduction. These include features such as guided meditations, breathing exercises, visualization techniques, and progressive muscle relaxation.

- Education and information

  Several mental health apps offer educational content and information on various mental health topics. These resources aim to increase students' knowledge about mental health, common challenges, coping strategies, and available support options.

- Crisis support and emergency resources

  Some mental health apps include information about helplines, crisis resources, and emergency services. These features ensure that students have access to immediate assistance in times of crisis or when they need urgent support.

ReachOut WorryTime

ReachOut Australia provides a secure and private platform for young individuals to freely express themselves, gain deeper insights into their lives, connect with non-judgmental supporters, and develop the resilience necessary to overcome challenges both presently and in the future. Created exclusively for and with young people in mind, ReachOut is a fully digital platform that ensures anonymity and confidentiality. It empowers young individuals to connect on their own terms, offering various support options such as one-on-one peer assistance, moderated online communities, valuable tips, inspiring stories, and a wealth of resources. This comprehensive range of support allows young people to engage in a manner that suits their preferences and schedules.

In regard to addressing mental health issues or difficulties in students, the ReachOut creators have dedicated a separate section where users can learn more about different topics that play an important role in a student's life, such as bullying, study stress, or social media. This section is also useful for educators who want to improve their ways of teaching by introducing to the classroom different activities or topics of discussion.

MindDoc

The MindDoc Companion is an app designed for individuals who wish to monitor and manage their mental health effectively. By tracking moods, symptoms, circumstances, and challenges, users can identify patterns in their mental well-being. As time progresses, the app offers

automated feedback and personalized recommendations. It provides insights not only into specific symptoms, issues, and personal strengths but also into the overall state of a user's emotional well-being. After 14 days, the app offers a recommendation on whether it would be beneficial for users to seek consultation with a mental health professional for further diagnosis and treatment.

The MindDoc Companion is not an application designed specifically for students but for anyone who feels the need to tune into their emotional needs or wants to learn more about how to improve their mental health.

Clinical Partners

Clinical Partners is recognized as a prominent provider of mental health services in the UK. They offer assistance to individuals and families facing mental and emotional challenges, working in collaboration with both private entities and the NHS (National Health Service). Their primary objective is to ensure that people receive the necessary support and resources to address their mental health needs effectively.

They offer direct support for children and adults, either from their own team of professionals or by connecting the person in need with a specialist in the issued field. They also provide a separate section dedicated to a wide range of online tests that give an evaluation regarding different mental health disorders, such as PTSD (post-traumatic stress disorder), depression, anxiety, or ADHD (attention deficit hyperactivity disorder).

Chapter 3

Practical application – Mental Health Checker for Students

3.1 Concept

Mental Health Checker for Students represents the practical application that makes the object of the thesis. It is a web application whose usability and functionality revolve around a mental health quiz that analyzes how the academic environment can impact students' daily activities. Through my research and application of my field-related knowledge, I have created a powerful tool for educators and mental health professionals to better understand and support students in achieving their full potential.

3.2 Application description

The application comes with several functionalities, all presented in a user-friendly manner. Given that the main scope of the application is the mental health quiz, which can later redirect the user to other available resources or functionalities, such as quiz analytics and performance overview, the secondary group of features stated by the application accommodates user management, providing the ability to register, log into the account, and track the mental health progress.

However, the concept of the application is that the student can use the safety of anonymity to expose his vulnerabilities related to the academic context. Given this, the features responsible for user management were built so that the completion of the quiz is in no way dependent on or related to registration within the application.

When entering the web site, the user is welcomed with the landing page of the application, and it gives the user the possibility to be redirected to all the available resources within the app, either by using the navigation bar or from the slideshow containing a presentation of the functionalities.

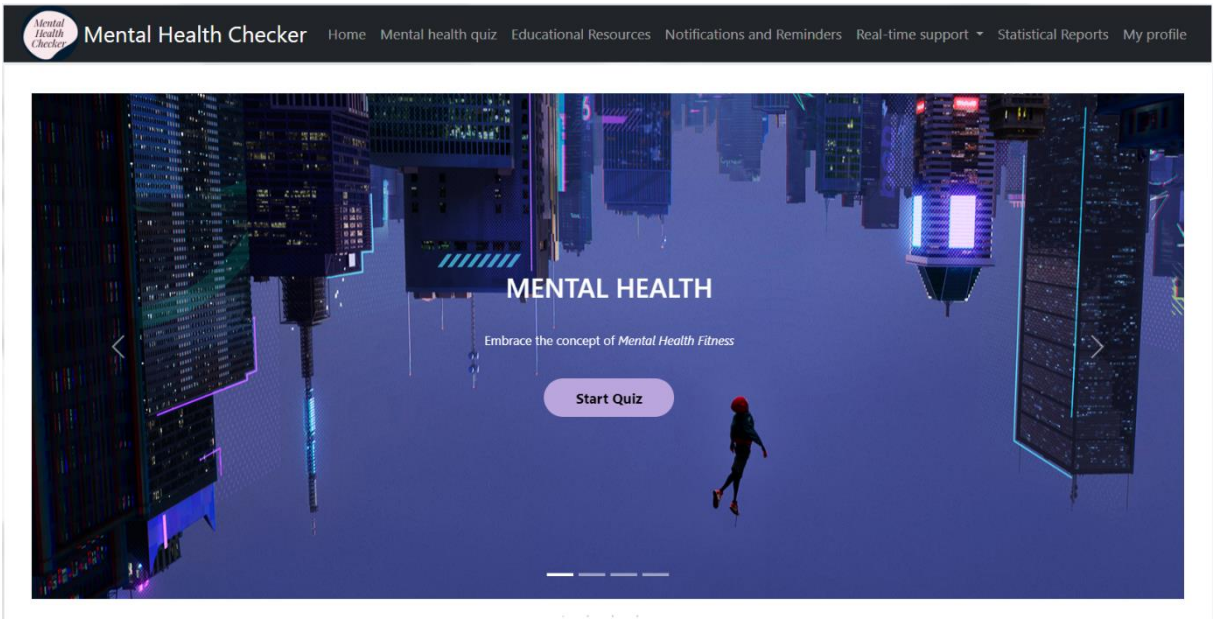

Figure 3.2.1: Landing page of the application

Through the completion of the quiz by users, the aim of investigating the extent to which mental health problems are associated with academic functioning can be met. Additionally, the application provides detailed analytics on user activity, including quiz completion rates, quiz scores, and user engagement, allowing users to track their progress in improving their mental health state and subscribe to a newsletter that provides regular self-care tips via mail. With personalized user profiles, individuals can view their most recent quiz score, update subscription preferences or personal information, and monitor their progress over time. Moreover, the administrator user role is available for managing user accounts and newsletter distribution, with the admin being able to send the user the weekly or monthly newsletter (depending on the user's preference) or delete the user's account.

3.2.1 Process of creating the quiz

When preparing the dataset for creating the mental health quiz for the web application, several important considerations were taken into account, such as legitimate research, the topics' diversity, reliability, and adaptability.

The questions included in the mental health quiz were carefully curated based on established research and with the help of a professional, a certified psychologist with over 15 years of work experience. This ensures the validity and reliability of the quiz in measuring relevant aspects of mental health within the academic context. The questions cover a wide variety of topics related to mental health, including stress, anxiety, depression, time management, coping mechanisms, and social support. By including a diverse set of topics, the quiz aims to provide a thorough assessment of a student's mental well-being within the academic environment. The dataset was created with adaptability and scalability in mind. It can be easily modified or expanded to accommodate future updates, improvements, or customization based on feedback or emerging research in the field of mental health.

Regarding the sensitivity of the questions and the privacy of the users taking the quiz, steps were taken to ensure that the questions were respectful and non-intrusive, and along with the option of taking the quiz without registration, the confidentiality and anonymity of the participants were protected.

Questions dataset:

- About how often did academic activities (courses, laboratories, exams) make you feel stressed (tense, irritable)?
- About how often did you feel that your effectiveness in learning is affected by the pressure that is placed on you?
- About how often was your sleep schedule affected by stress during a year of study that involves attending classes and laboratories?
- About how often have you struggled with anxiety when you had to present your work in front of your peers or teachers?
- About how often did you fail to meet the deadlines for handing in assignments?
- About how often do you think you have not been able to maintain a healthy relationship with your family because of the academic environment?
- About how often did you regret your choice to attend college?

- About how often did you have a positive attitude when encouraging others to attend college?
- About how often did you feel that everything was an effort?
- About how often did you feel panicked or overwhelmed by things in your life?

3.2.2 Answer scheme

The answer scheme used in the quiz consists of the following options:

- None of the time
- A little of the time
- Some of the time
- Most of the time
- All of the time

The five gradations offer a range of options that cover a spectrum from no occurrence to a complete occurrence, creating a 4-scale questionnaire and providing a detailed framework for assessing the impact of the academic context on various aspects of mental health. Given that this scheme is flexible and adaptable to different scenarios or contexts within the academic environment, it can also be easily modified or expanded upon to accommodate specific needs or changes in the field.

3.3 Used Technologies

3.3.1 C#

C# is a powerful, statically typed programming language that serves as the backbone of different frameworks for building web applications, enabling developers to write efficient and maintainable code. Often the programming language of choice for ASP.NET Core MVC, C# combines the power of object-oriented programming with the simplicity and expressiveness of

modern programming languages. It offers a rich set of features, such as strong typing, garbage collection, and extensive libraries, enabling developers to write clean, efficient, and scalable code. Additionally, C# is designed to promote productivity and maintainability by emphasizing code readability and reducing common programming errors.

3.3.2 ASP.NET Core MVC

ASP.NET Core MVC is a framework that follows the Model-View-Controller architectural pattern, providing a structured approach for building robust and scalable web applications. This framework embraces the principles of separation of concerns, allowing developers to divide the application logic into distinct components. The Model represents the data and business logic, the View handles the presentation layer, and the Controller manages the flow of information between the Model and the View. This separation promotes code organization, modularity, and testability, resulting in more manageable and maintainable codebases. Nevertheless, ASP.NET Core MVC leverages the cross-platform and open-source .NET Core framework, which provides a runtime environment for executing applications. .NET Core offers high performance, scalability, and cross-platform compatibility, allowing developers to target different operating systems, including Windows, macOS, and Linux.

3.3.3 Angular

Angular is a widely adopted frontend framework for building dynamic and responsive web applications, following a component-based architecture. It uses TypeScript as its programming language while providing a comprehensive set of tools and features for structuring the frontend logic, managing data binding, handling routing, and creating reusable UI components. Given that Angular offers a robust framework for developing modern, scalable, and maintainable frontend applications, a user-friendly and visually appealing interface has been created, making use of the architecture that allows for modular and reusable code.

3.3.4 Bootstrap

Bootstrap is a popular CSS framework that provides a collection of pre-designed components, layouts, and styling utilities. It simplifies the process of creating visually appealing and responsive web pages by offering a grid system, responsive breakpoints, and a wide range of customizable components. Bootstrap ensures consistency and ease of development across different screen sizes and devices, making it a valuable tool for frontend development. Additionally, Bootstrap can help in creating 'responsible' applications, i.e., regarding the size of the window, the layout of the page is not affected, meaning that all elements remain fully functional and accessible.

3.3.5 Microsoft SQL Server Management Studio

Microsoft SQL Server Management Studio (SSMS) is a powerful and comprehensive integrated development environment (IDE) used for managing, configuring, and administering Microsoft SQL Server. It provides a graphical user interface (GUI) that allows database administrators, developers, and other users to interact with SQL Server databases efficiently.

3.4 Architecture and Design

The backend and frontend concerns of the application have been separated, making it possible to achieve a decoupled architecture that allows for flexibility and scalability.

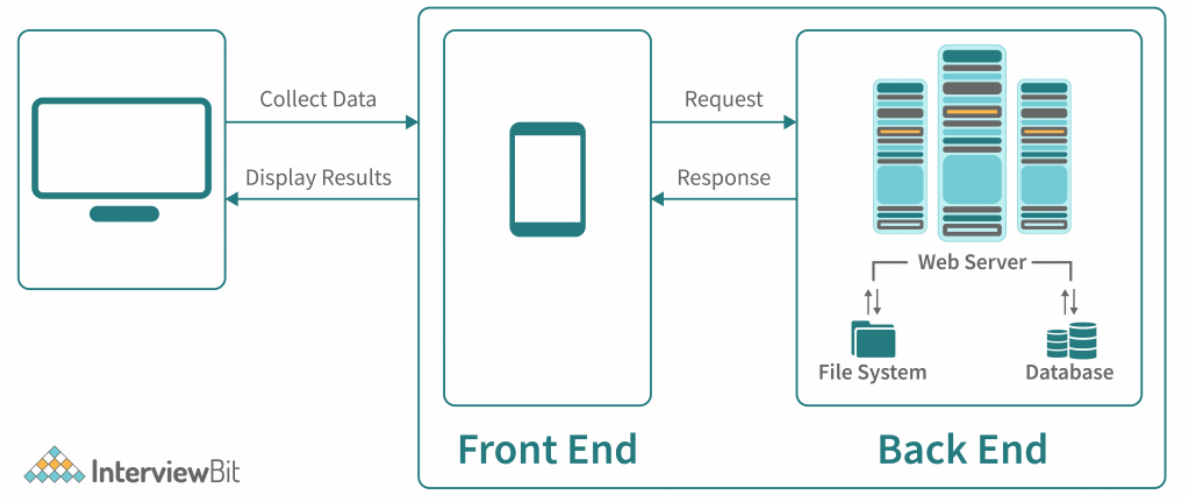

Figure 3.4.1: Web application architecture [18]

3.4.1 Database structure

An application's database structure is a vital component that determines data organization, storage, and retrieval. It serves as the foundation for data management and is critical to the application's general functionality and performance. In this section, we will dig into the complexities of **Mental Health Checker**'s database structure, offering an overview of its architecture, entities, connections, and essential features.

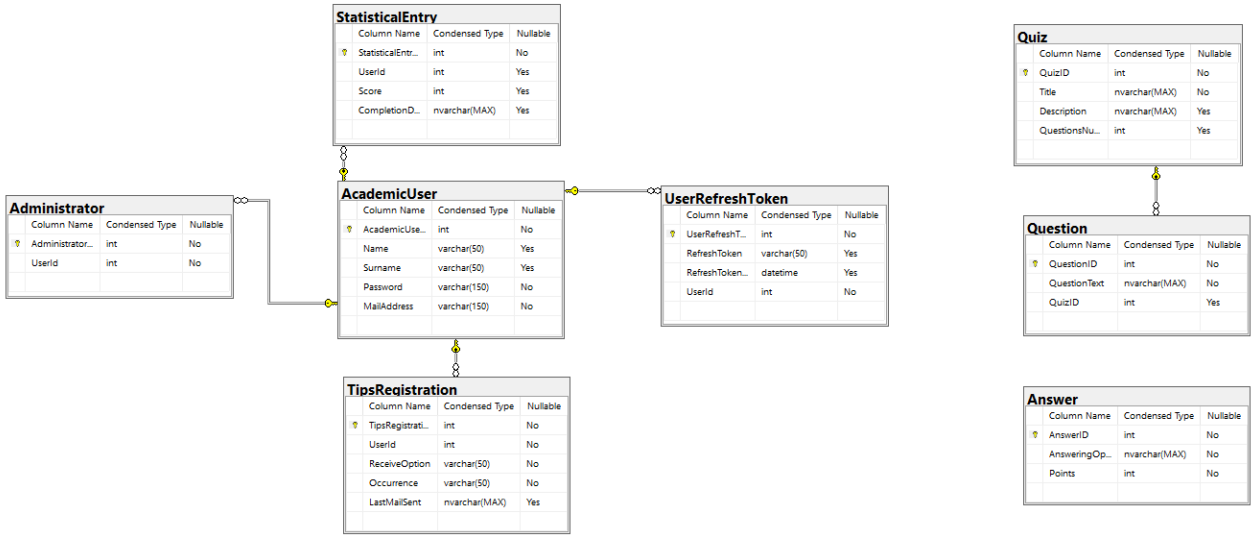

Figure 3.4.2: Database structure diagram

Within the database structure, various tables have been defined that correspond to the entities within the application domain: the registered users and the token information relevant to their log in sessions, the questions of the mental health quiz, each user's result after completing the quiz, and the details regarding the user's newsletter subscription. Additionally, relationships between the tables were established, such as one-to-many, to accurately represent the associations between the entities (as can be seen in Figure 3.4.2).

3.4.2 Backend architecture

The backend architecture of the application is built on the ASP.NET Core MVC framework with C# as the programming language, following the Model-View-Controller (MVC) pattern.

The Model component is responsible for managing the application's data and interacting with the database. The backend communicates with the database, whose structure is illustrated in Figure 3.4.2, using Entity Framework Core, an Object-Relational Mapping (ORM) tool that simplifies database interactions and enhances productivity, handling tasks such as data validation, retrieval, and manipulation.

```
protected override void OnConfiguring(DbContextOptionsBuilder optionsBuilder)
{
    optionsBuilder.UseSqlServer("Data Source=localhost;Initial Catalog=MentalHealthCheckerDB;" +
                                "Integrated Security=True; trusted_connection=true;encrypt=false");
}
```

Figure 3.4.3: Configuring the Database Context Builder

```
protected override void OnModelCreating(ModelBuilder modelBuilder)
{
    modelBuilder.Entity<AcademicUser>(entity =>
    {
        entity.HasKey(e => e.AcademicUserId).HasName("PK__Academic__C519E569A8CB95B5");

        entity.ToTable("AcademicUser");

        entity.HasIndex(e => e.MailAddress, "UQ__Academic__F58A94FCE606C021").IsUnique();

        entity.Property(e => e.AcademicUserId).HasColumnName("AcademicUserID");
        entity.Property(e => e.MailAddress)
            .HasMaxLength(150)
            .IsUnicode(false);
        entity.Property(e => e.Name)
            .HasMaxLength(50)
            .IsUnicode(false);
        entity.Property(e => e.Password)
            .HasMaxLength(150)
            .IsUnicode(false);
        entity.Property(e => e.Surname)
            .HasMaxLength(50)
            .IsUnicode(false);
    });
```

Figure 3.4.4: Configuring the User model and defining the database schema.

```
public partial class AcademicUser
{
    1 reference
    public AcademicUser()
    {
        UserRefreshTokens = new HashSet<UserRefreshToken>();
        StatisticalEntries = new HashSet<StatisticalEntry>();
        TipsRegistrations = new HashSet<TipsRegistration>();
        Administrators = new HashSet<Administrator>();
    }

    8 references
    public int AcademicUserId { get; set; }
    2 references
    public string? Name { get; set; } = null!;
    2 references
    public string? Surname { get; set; } = null!;
    5 references
    public string Password { get; set; } = null!;
    8 references
    public string MailAddress { get; set; } = null!;
    2 references
    public virtual ICollection<UserRefreshToken> UserRefreshTokens { get; set; }
    2 references
    public virtual ICollection<StatisticalEntry> StatisticalEntries { get; set; }
    2 references
    public virtual ICollection<TipsRegistration> TipsRegistrations { get; set; }
    2 references
    public virtual ICollection<Administrator> Administrators { get; set; }
}
```

Figure 3.4.5: The AcademicUser Model used in the backend architecture

The Controller component of the backend part handles the business logic of the application, including processing user requests, managing authentication and authorization, and orchestrating the flow of data between the frontend and backend. Given that this application does not use Razor view or server-side rendering for the user interface but an Angular frontend, the backend APIs (Application Programming Interfaces) exposed by the Controller component guarantee communication in regards to retrieving data, updating data, or submitting quiz responses.

```
[ApiController]
[Route("[controller]")]
1 reference
public class UserController : ControllerBase
{
    private readonly IUserService _userService;

    0 references
    public UserController(IUserService userService)
    {
        _userService = userService;
    }

    [HttpGet("get-all-users")]
    0 references
    public async Task<ActionResult<List<AcademicUser>>> GetAllUsers()
    {
        return await _userService.GetAllUsers();
    }
```

Figure 3.4.6: ControllerBase inheritance example

Controllers in ASP.NET Core MVC that inherit from ControllerBase serve as the backbone of your application's logic and are responsible for handling user requests and coordinating the flow of data. ControllerBase provides a set of base functionalities and features that make it easier to develop and manage controllers, such as routing options, request handling, or dependency injection.

Authentication mechanism

```
builder.Services.AddAuthentication(opt =>
{
    opt.DefaultAuthenticateScheme = JwtBearerDefaults.AuthenticationScheme;
    opt.DefaultChallengeScheme = JwtBearerDefaults.AuthenticationScheme;
}).AddJwtBearer(options =>
{
    options.TokenValidationParameters = new TokenValidationParameters
    {
        ValidateIssuer = true,
        ValidateAudience = true,
        ValidateLifetime = true,
        ValidateIssuerSigningKey = true,
        ValidIssuer = builder.Configuration.GetSection("JWTSetup")["Issuer"],
        ValidAudience = builder.Configuration.GetSection("JWTSetup")["Audience"],
        IssuerSigningKey = new SymmetricSecurityKey(Encoding.UTF8.GetBytes(builder.Configuration.GetSection("JWTSetup")["Key"]))
    };
});
```

Figure 3.4.7: Using JwtBearerDefaults package as the default authentication scheme

To ensure security, the application uses JSON Web Tokens (JWT) as the authentication mechanism in the backend architecture. The integration of such a mechanism consists of:

- Generating a JWT containing relevant user information and signing it using a secret key upon a successful authentication, by using the JwtBearerDefault package that handles the token creation process;

```
public string GenerateAccessToken(IEnumerable<Claim> claims)
{
    var secretKey = new SymmetricSecurityKey(Encoding.UTF8.GetBytes(_config.Key));
    var signinCredentials = new SigningCredentials(secretKey, SecurityAlgorithms.HmacSha256);
    var tokenOptions = new JwtSecurityToken(
        issuer: _config.Issuer,
        audience: _config.Audience,
        claims: claims,
        expires: DateTime.Now.AddMinutes(2),
        signingCredentials: signinCredentials
    );
    var tokenString = new JwtSecurityTokenHandler().WriteToken(tokenOptions);
    return tokenString;
}
```

Figure 3.4.8: Generation of an access token

```
public string GenerateRefreshToken()
{
    var randomNumber = new byte[32];

    using (var rng = RandomNumberGenerator.Create())
    {
        rng.GetBytes(randomNumber);
        return Convert.ToBase64String(randomNumber);
    }
}
```

Figure 3.4.9: Generation of a refresh token

- Transmitting the JWT to the client (in our case, the Angular frontend) once it is generated;
- Validating requests with the backend's authentication middleware, indicating unauthorized access in the case of an invalid or missing JWT;

```
public ClaimsPrincipal GetPrincipalClaimFromExpiredToken(string token)
{
    var tokenValidationParameters = new TokenValidationParameters
    {
        ValidateIssuer = false,
        ValidateAudience = false,
        ValidateLifetime = false,
        ValidateIssuerSigningKey = true,
        ValidIssuer = _config.Issuer,
        ValidAudience = _config.Audience,
        IssuerSigningKey = new SymmetricSecurityKey(Encoding.UTF8.GetBytes(_config.Key)),
    };
    var tokenHandler = new JwtSecurityTokenHandler();
    var principal = tokenHandler.ValidateToken(token, tokenValidationParameters, out SecurityToken securityToken);
    var jwtSecurityToken = securityToken as JwtSecurityToken;

    if (jwtSecurityToken == null ||
        !jwtSecurityToken.Header.Alg.Equals(SecurityAlgorithms.HmacSha256, StringComparison.InvariantCultureIgnoreCase))
        throw new SecurityTokenException("Invalid token");

    return principal;
}
```

Figure 3.4.10: Validating the token within the Authentication Middleware

- Assuring authorization and access control to specific resources or functionalities within the application based on the claims extracted from the validated JWT.

In order to ensure the security of user accounts, a hash class was implemented within the backend architecture of the application. This class employs the widely-used MD5 hashing algorithm to transform plain-text passwords into a fixed-length hash representation before storing them in the database.

```
public static class Hasher
{
    2 references
    public static string CreateMD5(string input)
    {
        using (MD5 md5 = MD5.Create())
        {
            if (input != null)
            {
                byte[] inputBytes = System.Text.Encoding.ASCII.GetBytes(input);
                byte[] hashBytes = md5.ComputeHash(inputBytes);
                return Convert.ToHexString(hashBytes);
            }
            else
                return "Empty input";
        }
    }
}
```

Figure 3.4.11: MD5 hashing for password storage

3.4.3 Frontend architecture

The frontend architecture of the application is designed using a combination of Bootstrap and Angular frameworks while following a component-based approach. Each functional aspect of the application is encapsulated within individual components, which are reusable and independent entities. These components communicate with each other through defined inputs and outputs, facilitating a modular and loosely coupled design. When creating a new component, four other files are generated, each representing a different functional aspect of the component itself:

- HTML file containing the HTML code of the page;
- CSS file responsible for the style;
- TS file containing the TypeScript code;
- MODULE TS file containing code for routing, lazy loading, or forms.

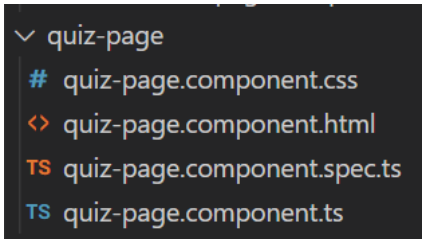

Figure 3.4.12: Structure of a component

The navigation within the application is handled by Angular's routing system. Furthermore, in order to handle the data collection and validation within the application, Angular's forms module has been used. This module provides powerful features for form handling, including form controls, form groups, and form validation, ensuring accurate and reliable data submission.

The inclusion of Bootstrap's CSS into the Angular specific scheme allows efficient usage of Bootstrap's styling and interactive elements, while Angular's component-based structure simplifies the integration process.

Hypertext Transfer Protocol

The interaction between the web client and the web server occurs via HTTP requests. HTTP, or Hypertext Transfer Protocol, is a simple protocol that facilitates the communication and transfer of data between web clients and servers on the internet.

The two main types of HTTP requests are:
1. GET – used to retrieve data or resources from the web server;
2. POST – used to submit data to the web server, typically to create or update resources on the server;

Additionally, there are other types of HTTP requests, such as:

- PUT – used to update or replace a resource on the server;
- DELETE – used to remove a resource from the server;
- HEAD – similar to a GET request but only retrieves the headers of a response;
- PATCH – used to partially update a resource on the server;
- OPTIONS – used to retrieve the available communication options supported by the server.

3.5 Development process

As mentioned previously in 3.2, the application offers a range of user-friendly functionalities that are easily accessible. Besides the user management capabilities that enable registration and login for the user, the application's primary focus is the mental health quiz, which not only assesses the user's mental health but also provides links to additional resources and functionalities. These include features like quiz analytics and performance overview, allowing users to track their progress.

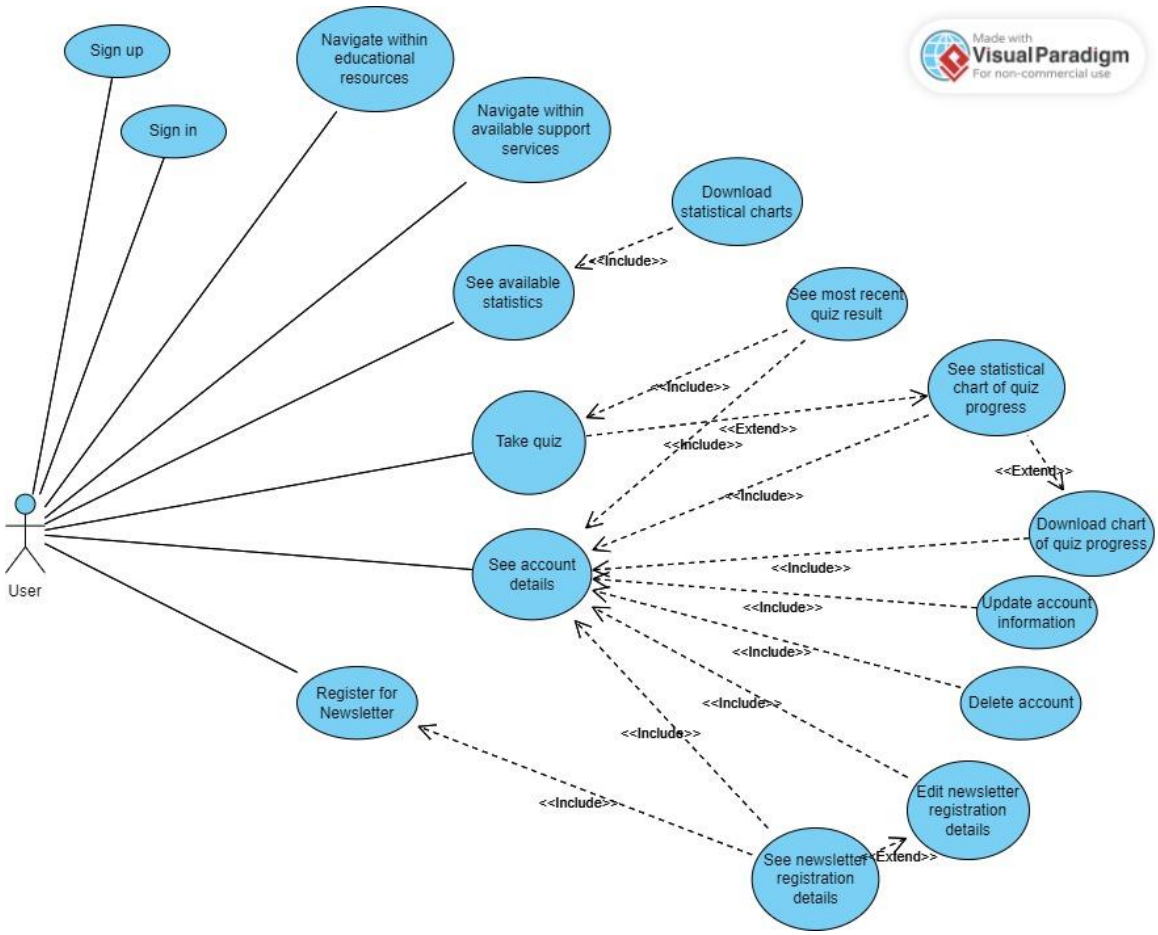

Figure 3.5.1: User use cases diagram

Quiz Completion

The certified psychologist that helped me throughout the process of creating the quiz has approved the policy of assigning to each option from the Answer Scheme mentioned in 3.2 a number of points, making it possible to have a 4-scale questionnaire and cover a spectrum from `no occurrence` - associated with 10 points, to `complete occurrence` - 50 points. In this way, we were able to create a scoring system for the prognostic displayed at the end of the quiz, but with the acknowledgment that the displayed results are indicative only and do not form a formal diagnosis.

The indicative conclusions we drew are:

- If the final score is less than 21 points, then the user is unlikely to struggle with anxiety or depression;
- If the final score is greater than 21 points but less than 34 points, then the user may be experiencing symptoms of anxiety, depression, or stress related to the academic environment;
- If the final score is greater than 34 points, then the user is most likely to suffer from severe anxiety, depression, or other mental health disorders.

Quiz Analytics

In order to provide detailed analytics and reporting on user activity in regards to the mental health quiz and the application's functionalities, we utilized the Chart.js library, a powerful JavaScript library that provides a wide range of charting options. The web page within the application is specifically designed to display various statistics in a visually appealing and interactive manner. By incorporating Chart.js, it became possible to create informative and dynamic charts that effectively communicate to the user statistical data, such as:

- Quiz completion rates;
- The average score obtained on the quiz;
- Newsletter registration rates.

```
public partial class StatisticalEntry
{
    3 references
    public int StatisticalEntryId { get; set; }
    10 references
    public int? UserId { get; set; }
    11 references
    public int? Score { get; set; }
    2 references
    public string? CompletionDate { get; set; }
    1 reference
    public virtual AcademicUser User { get; set; } = null!;
}
```

Figure 3.5.2: Model representing the quiz completion of a user

Using the StatisticalEntry model (Figure 3.5.2) and related methods implemented in the service layer from the backend module, we were able to structure the data so that several user engagement reports could be created and displayed. Every quiz completion by a user is saved and stored in the database as an entry with the following properties: the identifier of the user who completed the quiz, the score obtained at the end of the quiz, and the date and time it was completed. By storing this information, several methods were implemented in the backend module, such as: computing the average score obtained on the quiz, computing the number of users that obtained a specific score, and obtaining all the entries associated with a user in order to display its progress.

Out of the available built-in chart types, we have used Line Chart and Donut Chart in order to display a graph containing the score obtained on the quiz and the date, and the number of users from most to least likely to suffer from mental health issues related to the academic context, respectively. Each chart has the option to be downloaded under the .JPG extension for further use.

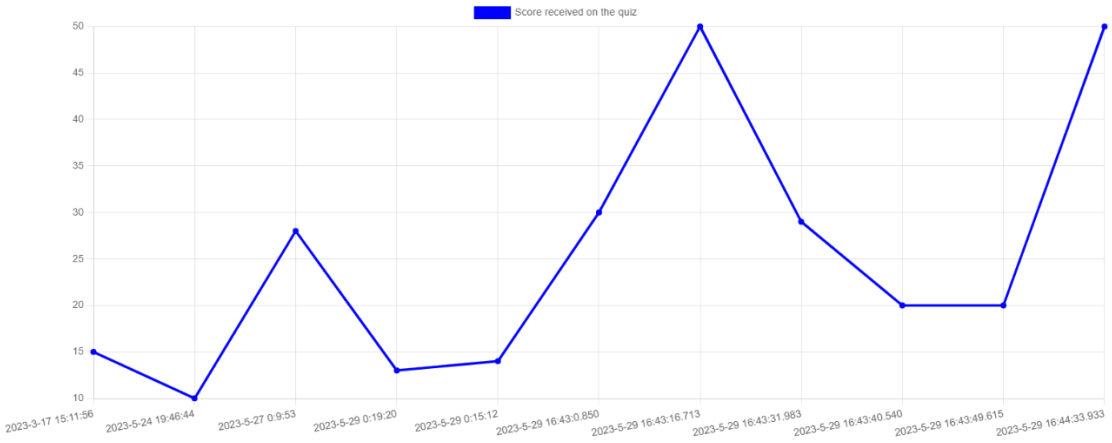

Figure 3.5.3: Example of a Line Chart embedded in the web application

Self-Care Newsletter Subscription

The users that consider they need assistance in dedicating time to self-care or need study or management tips due to their mental health struggle have the option to register for either a weekly or monthly newsletter sent via mail. The main aim of this functionality is to provide adequate and direct support to help students cope with academic challenges.

The registration details are saved with the help of a form that handles the input, and for the form creation FormGroup has been used, which is one of the four fundamental building blocks used to define forms in Angular along with FormControl, FormArray, and FormRecord. In order to improve overall data quality, there are Validator functions implemented for inputs that need a certain type of validation, such as mail addresses or phone numbers.

```
registrationForm = new FormGroup({
  inputOccurrence: new FormControl(""),
  inputMail: new FormControl("", [Validators.required,
    Validators.pattern("^[a-z0-9._%+-]+@[a-z0-9.-]+\\.[a-z]{2,4}$")]),
});
```

Figure 3.5.4: FormGroup used on the newsletter registration

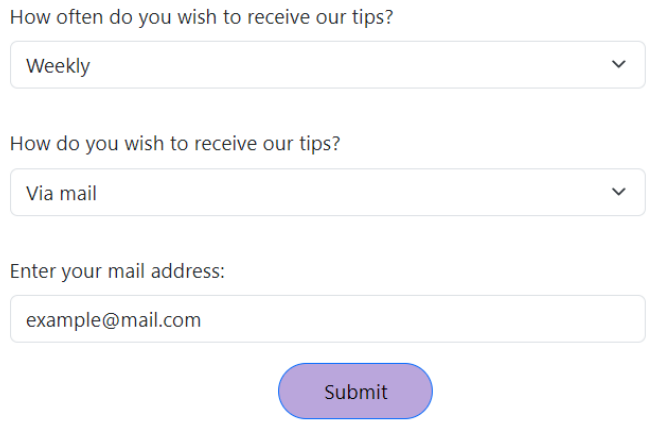

Figure 3.5.5: Form output for newsletter registration

SMTP (Simple Mail Transfer Protocol) Configuration for mailing

For the distribution of the newsletter via mail, the ASP.NET Core configuration services of the backend module were adapted to implement mailing within the application. In order to add the service layer for sending mail, two packages have been installed: *Mailkit* and *Mimekit*. After creating the necessary models for encapsulating the mail content (models for basic mail structure or for HTML templated mails), the newly installed packages were used to create an instance of *MimeMessage* and send it as a mail using a *SMTPClient* instance (specific to the *Mailkit* package) and Gmail as the SMTP Server.

```
public async Task SendEmailAsync(MailRequest mailReq)
{
    var email = new MimeMessage();
    email.Sender = MailboxAddress.Parse(_mailSettings.Mail);
    email.To.Add(MailboxAddress.Parse(mailReq.ToEmail));
    email.Subject = mailReq.Subject;

    var builder = new BodyBuilder();
    builder.HtmlBody = mailReq.Body;
    email.Body = builder.ToMessageBody();

    using var smtp = new SmtpClient();
    smtp.Connect(_mailSettings.Host, _mailSettings.Port, MailKit.Security.SecureSocketOptions.StartTls);
    smtp.Authenticate(_mailSettings.Mail, _mailSettings.Password);
    await smtp.SendAsync(email);
    smtp.Disconnect(true);
}
```

Figure 3.5.6: Method for sending mails using SMTP

User Performance Overview

Taking into account the previously mentioned functionalities, Quiz Completion, Analytics, and Newsletter Subscription, the structure of the user's personalized profile is adjusted according to the resources made available by those functionalities. Once a user has logged into his account, his profile page will display information regarding their most recent quiz score, newsletter subscription details, and mental health progress observed since account creation.

By using the service layer for manipulating the quiz completions associated with a user, a Line Chart containing all past quiz scores was created in the user's profile page with a downloading option, making it easier to track the progress and observe the mental health state changes. Besides visualizing the available information, the user also has the option to either retake the quiz, update the newsletter registration details, update the personal account information, or permanently delete the subscription.

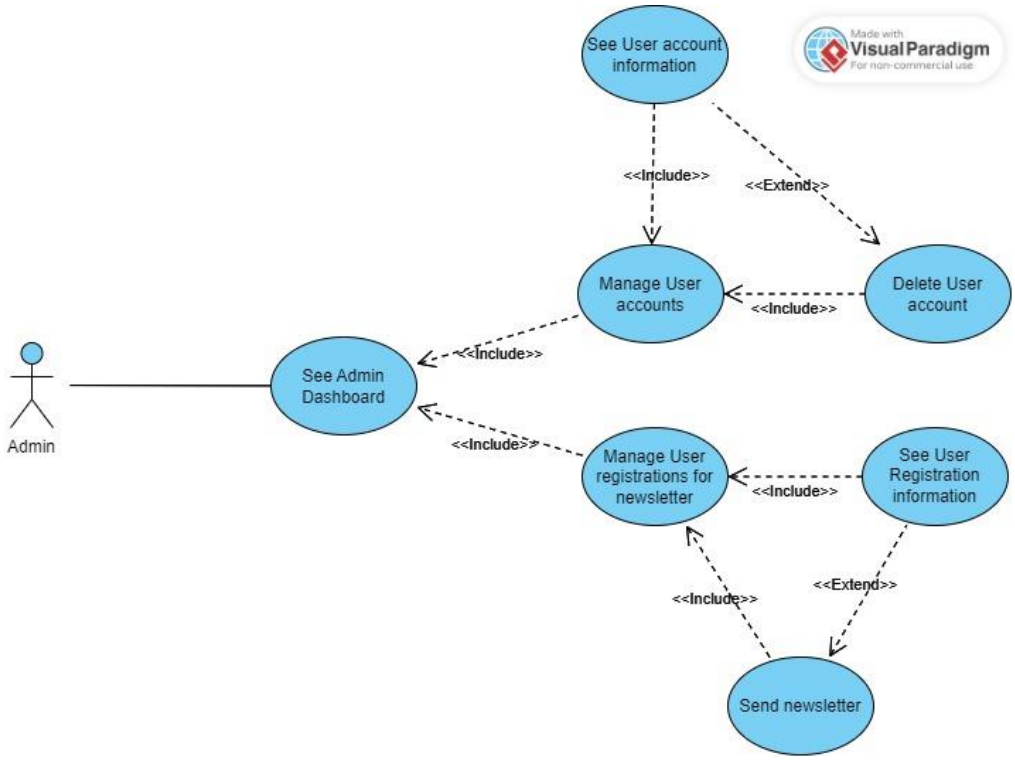

Figure 3.5.7: Administrator use cases diagram

Admin Dashboard

The second user role of the application is the administrator, with rights to manage user accounts and user subscriptions. The administrator is able to see all the registered accounts, along with personal information, and has the ability to permanently delete any user's account. Additionally, the administrator can manipulate the users newsletter registrations by seeing all the subscriptions and the specific details, including how often the user is supposed to receive the newsletter and when was the last time the newsletter was sent. If the user opted for a weekly newsletter and the last one was sent more than a week ago, then the administrator has the option to send the newsletter to the user directly from the Admin Dashboard (the same methodology for a monthly newsletter).

Educational Resources and Support Services

In addition to the user-interactive functionalities of the application, the user can also navigate through some educational resources, such as:

- → The impact of the academic context and how it can be dealt with;
- → Stress & time management, along with tips and recommendations;
- → Examples of study skills and how to incorporate them into a student's habits.

By browsing throughout the Support Services resource, the user can also benefit from redirection to platforms that offer direct contact and support, such as:

- → Emergency and support lines for mental health crises, bullying, violence, or vices;
- → Contact information for certified professionals and counselors.

3.6 Intelligent Data Analysis

3.6.1 Artificial Intelligence & Machine Learning

Artificial Intelligence

Artificial Intelligence (AI) refers to the development and deployment of computer systems that can perform tasks that would typically require human intelligence. It involves the creation of intelligent machines capable of learning, reasoning, problem-solving, and making decisions, often mimicking cognitive functions associated with human intelligence. AI systems are designed to analyze large amounts of data, recognize patterns, and adapt their behavior or improve their performance over time. The applications of AI are diverse and far-reaching. AI-powered systems can be found in areas such as healthcare, finance, transportation, manufacturing, and entertainment. They are used for tasks such as medical diagnosis, fraud detection, autonomous driving, quality control, language translation, virtual assistants, and recommendation systems.

Machine Learning

AI encompasses various subfields, including machine learning, natural language processing, computer vision, robotics, and expert systems. Machine learning is a core component of AI, enabling systems to automatically learn from data and improve their performance without being explicitly programmed. In 1959, Arthur Samuel [16] coined the term "Machine Learning" (ML) to describe a field of study that empowers computers to learn without explicit programming. There are four types of machine learning algorithms: supervised, semi-supervised, unsupervised, and reinforcement.

3.6.2 Artificial Neural Networks

Artificial neural networks (ANNs) consist of a vast number of interconnected simple processors, creating highly parallel systems. The fundamental building block of neural networks is the unit, also known as a node or neuron. A unit receives one or more inputs and applies a parameter, known as a weight, to each input. These weighted inputs are then summed along with a bias value, often set to 1. The resulting value is then passed through an activation function. The output is subsequently forwarded to other neurons in the neural network, if present, for further processing.

In his article, Luke Dormehl explains the structure of neural networks, which comprises an input layer, an output layer, and a hidden layer composed of units [17]. These units transform the input data into a format that the output layer uses to generate predictions.

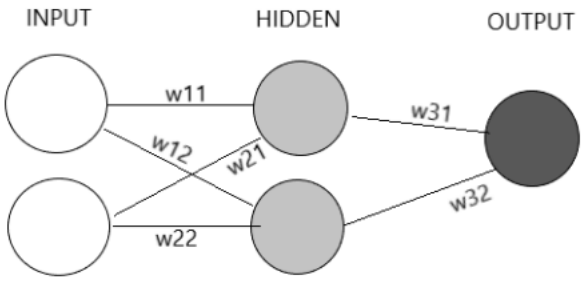

Figure 3.6.1: ANN with one hidden layer

Types of Neural Networks

There exist various types of neural networks, each with distinct applications and levels of complexity. Among them is the feedforward neural network – also called *multilayer perceptron*, which represents the most fundamental type where information flows in a unidirectional manner from input to output.

Another widely utilized neural network is the recurrent neural network, which is capable of bidirectional data flow. These networks possess enhanced learning capabilities and find extensive application in complex tasks like handwriting or language recognition.

Additional types include convolutional neural networks, Boltzmann machine networks, Hopfield networks, and several others. Selecting the appropriate network depends on the

available training data and the specific task at hand. In certain cases, a combination of approaches may be preferred, such as in challenging tasks like voice recognition.

3.6.3 Multilayer Perceptron

The origins of MLPs can be traced back to the 1950s, when psychologist Frank Rosenblatt introduced the perceptron, a single-layer neural network. However, the limitations of single-layer perceptrons became apparent, and they were unable to solve complex problems. Consequently, perceptrons lost popularity, and researchers shifted their focus to other machine learning algorithms, almost abandoning the idea of Neural Networks. In the 1980s, computer scientist Geoffrey Hinton proposed a new concept that reinvigorated interest in Neural Networks. He observed that the human brain consists of a complex network of interconnected neurons, not just a single neuron. Hinton hypothesized that the limitations of perceptrons could be overcome by stacking multiple layers of neurons to create a multi-layer neural network. This breakthrough led to the development of a powerful algorithm known as the "Multilayer perceptron (MLP)."

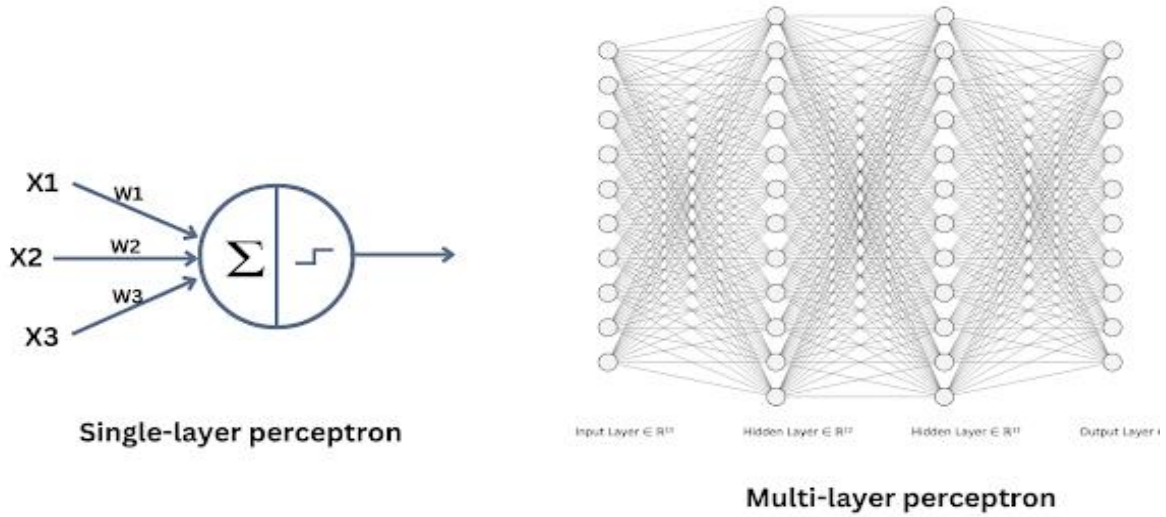

Figure 3.6.2: Single-layer perceptron vs Multi-layer perceptron [19]

As mentioned before, multilayer perceptrons form one of the types of neural networks. An MLP is an artificial neural network consisting of interconnected layers of "neurons." These artificial neurons are inspired by the neurons in the human brain and are designed to learn from complex data and make insightful predictions.

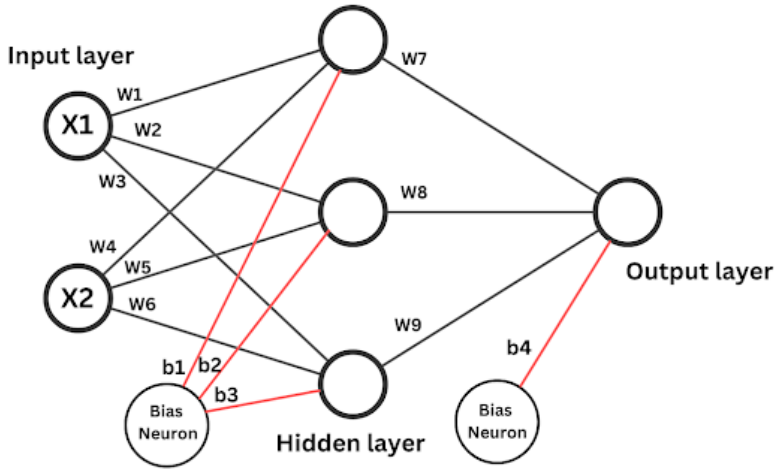

Figure 3.6.3: MLP weights and biases [19]

Within each of these neurons, we allocate weights and biases, which are considered the fundamental components of a Neural Network. Weights play a crucial role as they determine the intensity of the connections between neurons, essentially representing the "strength" of these connections. On the contrary, biases play a crucial role in determining the activation level of a neuron. They can be likened to a threshold that a neuron must surpass to generate an output. When the sum of the neuron's input and bias exceeds a specific value, it will generate an output. If not, it remains inactive. This flexibility enables the network to adapt to various input patterns and enhances its overall adaptability.

The computation of each neuron inside the network can be given by:

$$z = WX + bias$$

Figure 3.6.4: Neuron computation,

where W = weights of each neuron, X = inputs.

However, given that *z* is nothing but a linear model, introducing an Activation Function is needed so that the non-linearity of the MLP is guaranteed. An activation function is a non-linear mathematical function that determines whether neurons should be activated or not within a neural network. Its purpose is to introduce non-linearity into the network, allowing it to learn intricate patterns within the provided data.

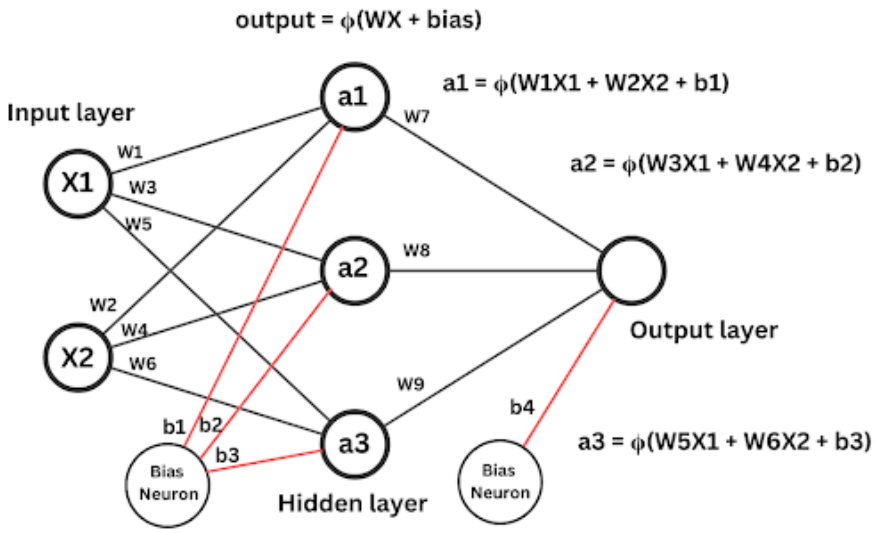

Figure 3.6.5: Weights, biases, and activation [19]

The output of each neuron can be computed with the help of the activation function in the following way:

$$output = \phi(WX + bias)$$

OR

$$z = WX + bias$$
$$output = \phi(z)$$

Figure 3.6.6: Output computation,

where **ϕ** is the activation function.

Rectified Linear Unit function

The ReLU (Rectified Linear Unit) function is a widely used activation function in neural networks. When applied, the ReLU activation function takes the weighted sum of inputs plus the bias (XW + bias). If the resulting value is greater than 0, it returns the same value as the output. On the other hand, if the value is less than or equal to 0, it outputs 0.

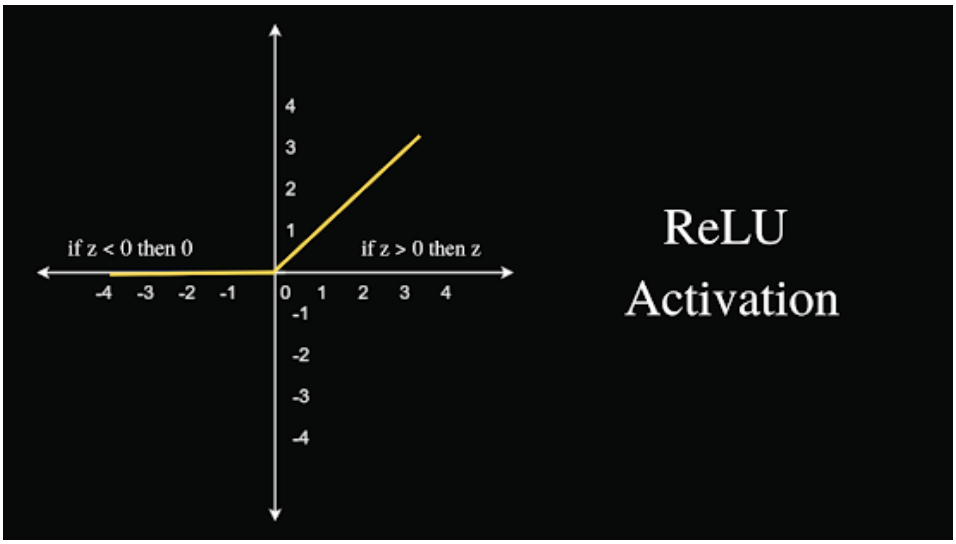

Figure 3.6.7: ReLU Activation function graph [19]

The mathematical expression of ReLU can be written as:

$$ReLU = max(0, z)$$

Figure 3.6.8: ReLU expression

MLP within Mental Health Checker

For an intelligent analysis of the data gathered with the help of the application, and more specifically, in order to obtain a prediction for the next score on the mental health quiz, we used the multilayer perceptron classifier model to manipulate the data extracted from the StatisticalEntry table from the database, which contains information about every quiz completion until the present.

The training data (the data extracted from the StatisticalEntry table) is propagated to the MLP through the input layers, then passes through the hidden layers. Next, by applying the activation function (in this specific case, the Rectified Linear Unit function), the output is generated at the output layer. The predicted output will then be compared to the actual output; hence error will be calculated.

In [20]:
```
import pandas as pd
import numpy as np

def lookup(date_pd_series, format=None):
    dates = {date:pd.to_datetime(date, format=format) for date in date_pd_series.unique()}
    return date_pd_series.map(dates)
```

In [21]:
```
entries = pd.read_csv('statEntry_export.csv', encoding="utf-16")
entries['CompletionDate'] = lookup(entries['CompletionDate'], format='%Y-%m-%d %H:%M:%S')
entries['timestamp'] = pd.to_datetime(entries['CompletionDate']).astype(np.int64)
```

In [22]:
```
# create a separate column containing  2 labels linked to the score
conditions = [(entries['Score'] < 30),
              (entries['Score'] >= 30)]
prognostics = ['slightly','considerably']
entries['MentalStateDeterioration'] = np.select(conditions, prognostics)
print(entries.drop('CompletionDate', axis=1).head())
```

```
   StatisticalEntryId  UserId  Score            timestamp  \
0                   1      20     40  1686409540901000000
1                   2      20     20  1686409562381000000
2                   3      20     40  1686409575139000000
3                   4      20     30  1686410015167000000
4                   5      20     50  1686410875578000000

  MentalStateDeterioration
0             considerably
1                 slightly
2             considerably
3             considerably
4             considerably
```

Figure 3.6.9: Multilayer Perceptron applied on the quiz completions – preparing the input data

In [39]:
```
X = entries.drop(['MentalStateDeterioration', 'CompletionDate'], axis = 1)
y = entries['MentalStateDeterioration']
print(X.head(2))
print(y.head(2))
```

```
   StatisticalEntryId  UserId  Score            timestamp
0                   1      20     40  1686409540901000000
1                   2      20     20  1686409562381000000
0    considerably
1        slightly
Name: MentalStateDeterioration, dtype: object
```

Figure 3.6.10: Preparing the datasets that will be used by the perceptron

Based on the test dataset, we were able to make a prediction regarding the evaluation of the next possible score on the quiz.

```
In [49]:
from sklearn.neural_network import MLPClassifier
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.4)

mlp = MLPClassifier(max_iter=500, activation='relu')
mlp.fit(X_train, y_train)

Out[49]:
MLPClassifier
MLPClassifier(max_iter=500)

In [50]:
pred = mlp.predict(X_test)
from sklearn.metrics import accuracy_score
accuracy_score(y_test, pred)

Out[50]:
0.8333333333333334
```

Figure 3.6.11: Model prediction and accuracy score

The model accuracy, precision, recall, and F1 score, also known as the harmonic mean of the precision and recall, were also assessed.

```
In [51]:
from sklearn.metrics import classification_report, confusion_matrix

confusion_matrix(y_test, pred)

Out[51]:
array([[5, 0],
       [1, 0]], dtype=int64)

In [54]:
print(classification_report(y_test,pred))

from sklearn.metrics import precision_score, accuracy_score, recall_score, f1_score
print(f"Accuracy: {round(accuracy_score(y_test, pred), 2)}")
# computed the following for 'considerably' label
print(f"Precision: {round(precision_score(y_test, pred, pos_label='considerably'), 2)}")
print(f"Recall: {round(recall_score(y_test, pred, pos_label='considerably'), 2)}")
print(f"F1_score: {round(f1_score(y_test, pred, pos_label='considerably'), 2)}")

              precision    recall  f1-score   support

considerably       0.83      1.00      0.91         5
    slightly       0.00      0.00      0.00         1

    accuracy                           0.83         6
   macro avg       0.42      0.50      0.45         6
weighted avg       0.69      0.83      0.76         6

Accuracy: 0.83
Precision: 0.83
Recall: 1.0
F1_score: 0.91
```

Figure 3.6.12: Statistical metrics

This example of intelligent data analysis that performs predictions can be a way to facilitate the interpretation or further development of the data generated within the application by mental health specialists.

Chapter 4

Conclusions

4.1 Further improvements

Given that the application's main focus is the mental health quiz designed for students, further development or use of the Mental Health Checker can include creating several quizzes targeting different aspects of mental health. Providing users with multiple self-assessment tests can be an additional step towards identifying struggles and getting the right help. In terms of adding more functionalities to the application, an online forum can also be created as a safe place for students to express their thoughts and concerns, with the possibility of receiving direct support from a professional. This would also include creating a new user role or assuming that behind the Administrator role is a certified psychologist.

Moreover, as we discussed in the previous chapter, the intelligent analysis of the collected data can be taken to an extent where specialists in the mental health field interpret it, taking into account the relevant improvements that can be made as a result of the analysis.

4.2 SWOT Analysis

A SWOT analysis is a strategic planning tool used to evaluate the strengths, weaknesses, opportunities, and threats of a business, project, or organization. By applying this analysis, we can determine how closely the Mental Health Checker is aligned with its success benchmarks. It also helps in understanding the competitive landscape as well as determining strategic directions for future growth.

Strengths of the Mental Health Checker

- It is addressed specifically to students;
- It provides a self-assessment process for identifying mental health struggles;
- It offers quick links and contact options to counselors, professionals, or emergency lines.

Weaknesses of the Mental Health Checker

- It does not address more specific mental health issues but rather a general mental health state;
- It does not provide direct support from a professional within the application, as in a forum;
- Every improvement made within the application that includes forms of prognostics or diagnoses must receive the approval of a certified psychologist.

Opportunities for the Mental Health Checker

- High growth and success rates given the underserved market for mental health applications designed specifically for students;
- Low number of competitors in Romania;
- Emerging need for addressing such issues and providing adequate support in the students' community.

Threats to the Mental Health Checker

- The mental health regulatory environment changing over time;
- The obligation to periodically adjust the application's resources according to students' needs and legal implementations;
- Adapting the mental health quiz to today's student issues and concerns.

Bibliography

[1] Duffy ME, Twenge JM, Joiner TE. Trends in Mood and Anxiety Symptoms and Suicide-Related Outcomes Among U.S. Undergraduates, 2007-2018: Evidence From Two National Surveys. J Adolesc Health. 2019;65(5):590-598.

[2] Arnett JJ. Emerging adulthood. A theory of development from the late teens through the twenties. Am Psychol. 2000;55(5):469-480.

[3] De Luca SM, Franklin C, Yueqi Y, Johnson S, Brownson C. The Relationship Between Suicide Ideation, Behavioral Health, and College Academic Performance. Community Ment Health J. 2016;52(5):534-540.

[4] Mansfield KE, Mathur R, Tazare J, et al. Indirect acute effects of the COVID-19 pandemic on physical and mental health in the UK: a population-based study. Lancet Digit Health. 2021;3(4):e217-e230.

[5] European University Association (EUA), The impact of Covid-19 on European higher education, https://eua.eu/resources/publications/989:the-impact-of-covid-19-on-european-higher-education.html.

[6] University of Bucharest, Starting school in a pandemic context survey, https://ires.ro/uploads/articole/ires_reinceperea-scolii_sondaj-de-opinie_ianuarie-2021.pdf.

[7] Gallagher Robert P. National Survey of Counseling Center Directors 2008. Project Report. The International Association of Counseling Services (IACS).

[8] Eurostat, How many students worked while studying in 2021?, https://ec.europa.eu/eurostat/web/products-eurostat-news/-/ddn-20220829-1.

[9] Murphy, M. C., & Archer, J. (1996). Stressors on the college campus: A comparison of 1985–1993. Journal of College Student Development, 37(1), 20–28.

[10] Kostanski M, Gullone E. Adolescent body image dissatisfaction: relationships with self-esteem, anxiety, and depression controlling for body mass. J Child Psychol Psychiatry. 1998;39(2):255-262.

[11] Teen Line, Invaluable College Mental Health Resources, https://teenlineonline.org/invaluable-college-mental-health-resources/.

[12] Barrera TL, Norton PJ. Quality of life impairment in generalized anxiety disorder, social phobia, and panic disorder. J Anxiety Disord. 2009 Dec;23(8):1086-90.

[13] Diener E, Diener M. Cross-cultural correlates of life satisfaction and self-esteem. J Pers Soc Psychol. 1995;68(4):653-663.

[14] Center for Collegiate Mental Health, 2022 Annual Report, https://ccmh.psu.edu/assets/docs/2022%20Annual%20Report.pdf.

[15] European University Association (EUA), Student mental health across Europe: towards a public mental health approach, https://eua.eu/resources/expert-voices/283-student-mental-health-across-europe-towards-a-public-mental-health-approach.html.

[16] William L. Hosch, Machine learning, https://www.britannica.com/technology/machine-learning.

[17] Luke Dormehl, What is an artificial neural network? Here's everything you need to know, https://www.digitaltrends.com/computing/what-is-an-artificial-neural-network/.

[18] InterviewBit, Web Application Architecture – Detailed Explanation, https://www.interviewbit.com/blog/web-application-architecture/.

[19] PyCodeMates, Multi-Layer Perceptron Explained: A Beginner's Guide, https://www.pycodemates.com/2023/01/multi-layer-perceptron-a-complete-overview.html.