HELWAN UNIVERSITY
Faculty of Computers and Artificial Intelligence
Computer Science Department

Autism Care System

A graduation project dissertation by:

[Moaz Ahmed Nageb (201900836)]

[Karim Ahmed Metwally (201900577)]

[Mahmoud Taha Ahmed (201900772)]

[Ahmed Khaled Ahmed (201900026)]

[Mariam Yehia Salah (201900806)]

[Neven Hamdy Saad (201900919)]

Submitted in partial fulfilment of the requirements for the degree of Bachelor of Science in Computers & Artificial Intelligence, at the Computer Science Department, the Faculty of Computers & Artificial Intelligence, Helwan University

Supervised by:

[*Dr. Ahmed Hesham*]

(June 2023)

Abstract

The Autism Care System project is a comprehensive web application that aims to provide a range of services to children with autism and their families. The app offers a variety of games that are specifically designed to help children with autism improve their cognitive, social, and communication skills. These games are developed by experts in the field of autism and are tailored to meet the unique needs of each child.

In addition to the games, the app also provides parents with a platform to find and book appointments with caregivers who specialize in working with children with autism. The app's caregiver directory includes detailed profiles of each caregiver, including their qualifications, experience, and availability.

One of the key features of the Autism Care System is its ability to detect if a child has autism. The app uses advanced machine learning algorithms to analyze a child's behavior and identify any signs of autism. This feature can be a valuable tool for parents who are concerned about their child's development and want to take early action to address any potential issues.

Overall, the Autism Care System is an essential tool for parents and caregivers who are working to support children with autism. Its range of features and services make it a valuable resource for anyone who is looking to improve the lives of children with autism.

Keywords

1. Autism spectrum disorder
2. Early intervention
3. Applied behavior analysis (ABA)
4. Speech therapy
5. Occupational therapy
6. Sens integration
7. Social skills training
8. Special education
9. Individualized education plan (IEP)
10. Assistive technology
11. Behavioral therapy
12. Parent training
13. Transition planning
14. Inclusion
15. Sensory-friendly environments.
16. MTCNN
17. MobileNet
18. Artificial Neural Networks

Acknowledgement

We would like to express our sincere gratitude to our supervisor, [Dr. Ahmed Hesham], for his invaluable guidance, support, and encouragement throughout this project. His expertise and insights have been instrumental in shaping the direction and scope of our Project, and we are deeply grateful for his unwavering commitment to our success.

We also would like to express our gratitude to our speech pathologist [Dr. Hadeer Yehia], for her valuable assistance and extensive knowledge in providing us with information and medical insights related to autism syndrome.

We would also like to thank our families and friends for their unwavering support and encouragement, especially during the challenging times. Their love and encouragement have been a constant source of motivation and inspiration, and we are truly blessed to have them in our life.

Finally, we would like to acknowledge the contributions of all the participants who generously gave their time and insights to this study. Without their participation, this project would not have been possible.

Thank you all for your support and encouragement.

Table of Contents

List of Abbreviations

- **ASD** => Autism Spectrum Disorder.
- **UI** => User Interface.
- **CJ** => Clinical Judgment.
- **DSM-IV** => Diagnostic and Statistical Manual fourth edition.
- **PDDs** => Pervasive Developmental Disorders.
- **ADI-R** => Autism Diagnostic Interview-Revised.
- **ADOS** => Autism Diagnostic Observation Schedule.
- **CARS** => Childhood Autism Rating Scale.
- **AQ** => Autism Spectrum Quotient.
- **CAST** => Childhood Asperger Syndrome Test.
- **M-CHAT** => Modified Checklist for Autism in Toddlers.
- **rs-FMRI** => resting state functional magnetic resonance imaging.
- **CNN** => Convolution Neural Networks.
- **AUC** => Area Under ROC curve
- **ROC** => Receiver Operating Characteristic curve
- **NPV** => Negative Predictive Value
- **KNN** => k-Nearest Neighbours
- **SVM** => Support Vector Machine
- **CBI** => Computer-Based Instruction
- **BYOB** => Build Your Own Block
- **ABA** => Applied Behaviour Analysis
- FER => Face emotion recognition
- **AI** => Artificial Intelligence
- **FACS** => Facial Action Coding System
- **LBP** => Local Binary Patterns
- **HOG** => Histogram of Oriented Gradients
- **ROI** => Region of Interest
- **MTCNN** => Multi-task Cascade Convolution Neural Network
- **LSTM** => Long Short- Term Memory
- **SAEs** => Sparse Autoencoders
- **VAEs** => Variational Autoencoders

Chapter 1

[Introduction]

Chapter 1 provides an overview of the autism assistant app, which is designed to help children with autism improve their cognitive, social, and communication skills through engaging games. The app also includes a booking system for parents to schedule appointments with caregivers and a screening tool to detect if their child may have autism. The problem statement highlights the challenges faced by families in accessing affordable and accessible resources for their children with autism. The scope and objectives of the app are also outlined, which aim to provide an affordable and accessible option for families to support their child's development.

1.1 Overview

The autism care system is designed to help children with autism and kids with special needs improve their abilities by playing games and provide parents with a platform to find and book appointments with caregivers and detect if their children have autism.
The app will include a variety of games that are specifically designed to help children with autism improve their cognitive, social, and communication skills.
These games will have multiple levels to be tailored to different age groups and skill levels, ensuring that children can progress at their own pace, encouraging the children to proceed by using the reinforcement technique applied in games and the token economy system.
Parents will be able to create a profile for their child to see his/her progress in the Games.

The app will also feature a booking system for parents to schedule appointments with caregivers, such as therapists or psychologists. Parents will be able to search for caregivers based on their location, specialty and rating the app will also include a screening tool to help parents detect if their child may have autism.
In addition to the features mentioned, it is important to note that the sessions with caregivers, such as therapists or psychologists, that help children with autism improve their skills can be highly expensive. This makes it challenging for many families to access the necessary care for their children.

Furthermore, the process of screening for autism can be complicated and expensive, requiring specialized training and tools. The screening tool provided by the app can save parents time and money by providing an initial assessment of their child's symptoms.
The autism assistant app can help bridge this gap by providing a more affordable and accessible option for families to support their children's development. The app's games are designed to be engaging and fun while also helping children with autism improve their cognitive, social, and communication skills.

Overall, the autism care system can be a valuable resource for families who may not have access to the expensive resources necessary to support their child's development.
It offers a more accessible and affordable option to help children with autism improve their abilities and reach their full potential.

1.2 problem statement

According to the Autism Speaks organization that 1 in 44 children is diagnosed with autism in the United States, Statistics in Egypt shows that 1 in 160 is diagnosed with autism Many families with children with autism face barriers in accessing affordable and accessible resources to support their child's development.
The sessions with caregivers that help children improve their skills can be highly expensive, and the process of screening for autism can be complicated and expensive.
This can make it challenging for families to provide the necessary care for their children with autism.

And finding specialists who are experienced in working with children with autism can be challenging, especially in areas with limited access to healthcare resources. This can leave families feeling isolated and without adequate support to manage the challenges of raising a child with autism.

Furthermore, the lack of an active community supporting families with children with autism can compound the challenges faced by these families. Without a supportive community, parents may struggle to find information and resources, feel overwhelmed and stressed, and struggle to navigate the complex landscape of care options for their child.

The autism care system aims to address these barriers by providing a more affordable and accessible option for families to support their child's development through engaging and fun games and by providing a screening tool and a booking system for caregivers

1.3 scope and objectives

The system aims to provide an affordable and accessible option for families with children with autism to support their child's development. The app will include a variety of games that are specifically designed to help children with autism improve their cognitive, social, and communication skills. These games will have multiple levels to be tailored to different age groups and skill levels, ensuring that children can progress at their own pace, and the app will use reinforcement techniques and a token economy system to encourage children to continue playing.

The app will also include a booking system for parents to schedule appointments with caregivers, such as therapists or psychologists, who specialize in working with children with autism. Parents will be able to search for caregivers based on their location and specialty. Additionally, the app will include a screening tool to help parents detect if their child may have autism.

The objectives of the autism assistant app are:

1. To provide an engaging and fun platform for children with autism to improve their cognitive, social, and communication skills.
2. To provide an affordable and accessible option for families with children with autism to support their child's development.
3. To offer a screening tool to help parents detect if their child may have autism and provide them with a report that they can share with their healthcare provider.
4. To create a platform for parents to easily find and book appointments with caregivers who specialize in working with children with autism.
5. To provide a resource for families to connect with a supportive community of other parents and caregivers who understand the challenges of raising a child with autism.

Overall, the autism care system aims to provide a comprehensive platform that addresses the various challenges that families with children with autism face in accessing affordable and accessible resources to support their child's development.

Chapter 2

[Literature Review]

In this chapter, our focus will be on examining applications that share similarities with our system. We will delve into the pros and cons of these comparable apps, providing a comprehensive understanding of their strengths and weaknesses. Additionally, we will offer a succinct background on pertinent theories and techniques that fall within the scope of our research. To ensure a well-rounded perspective, we will also conduct a literature review, highlighting the most recent and relevant papers in our field of study.

2.1 Similar Apps

2.1.1 Diagnosis

2.1.1.1 Clinical partners test for autism

- **Pros** : the UI is simple
- **Cons** : the website for mental health in general, it doesn't give any rating system for the specialists and the specialists only exists in the United Kingdom , there is no Arabic option for the test

2.1.2 Games

2.1.2.1 AutiSpark

- **Pros :** the website provide a complete plan to develop the child skills
- **cons :** the plan isn't for free.

2.1.2.2 Autism games

- **Pros :** the games are free.
- **Cons :** the UI for the website is not friendly nor simple and the games is so confusing for autistic children and hard to understand.

2.1.3 Booking system for caregivers

2.1.3.1 Koshofat

- **Pros :** the UI is friendly and simple
- **Cons:** The website is for booking a doctor in general not Specified to the autism or special abilities community and there are no specialists only doctors.

Conclusion:

" There is no website that combines all these features together which makes it difficult for families to find support and help they need to take care of their children."

2.2 Background

2.2.1 diagnosis

Autism symptoms are typically more noticeable and easier to identify in children between the ages of two and three. one in every 68 children has autism[1]. Therefore, medical professionals and psychiatrists worldwide have developed various screening methods to detect autistic traits in their early stages, allowing for prompt administration of necessary medications.

Diagnosing ASD can be a difficult task, given the variety of clinical techniques available, which often involve extensive observation and evaluation by licensed healthcare professionals [2,3].

The standard diagnostic procedures for ASD require medical professionals to conduct a clinical assessment of the patient's developmental age, which is based on several domains, such as communication, social skills, self-care, and behavioural excesses. This approach is commonly known as clinical judgment [4]. Until recently, the DSM-IV has been widely used by clinicians as the basis for diagnosing autistic behaviours [5], with autism being classified under the category of common PDDs.

Numerous methods, both clinical and self-screening, are available to evaluate individuals with ASD. Some of the well-known clinical methods include ADI-R, ADOS, CARS, Joseph Picture self-concept scale, and the social responsiveness scale [6,7]. These clinical methods are used for formal ASD diagnosis and treatment planning [8]. The effectiveness of some techniques, such as ADI-R and ADOS, has been clinically proven to distinguish autism from other related developmental disorders and to have sufficient validity and sensitivity [9]. However, some of these methods have been criticized for being time-consuming, having long questionnaires and scoring procedures, and requiring licensed clinicians and observers to administer them [10].

In addition to clinical diagnostic methods, there are also self-administered screening instruments created by various neuroscientists and psychologists in the autism and healthcare fields. These tools, including AQ, CAST, and M-CHAT. Most of these tools are based on clinical judgment methods and provide more user-friendly options for undergoing an ASD screening[11].

2.2.2 Games

2.2.2.1 Theory of applied behavioural analysis and its application in teaching.

The concept of teaching autistic children is not a recent development. In the 1960s, Ivar Lovaas introduced a teaching approach known as "applied behaviour analysis." This technique involves using environmental consequences to encourage or discourage specific behaviours. Objects, foods, and actions are used as rewards for desired behaviour, guided by a researcher (Lovaas, 1977; Charlop-Christy et al., 2002)[12,13]. Through numerous trials and sessions, children with autism gradually learn to interact predictably with people in their surroundings.

However, there are three primary limitations associated with this form of treatment:

- It necessitates multiple sessions with limited availability of trained professionals, which can impose financial strain on families.
- Teaching sessions demand focused attention and prolonged engagement from a practitioner or parent.
- The child is required to interact with a human being, yet one characteristic of ASD is a tendency towards anxious, detached, and solitary interactions with others (Baskett, 1996; Kanner, 1943)[14,15]. Therefore, relying on human interaction as the primary mode of teaching might present inherent challenges for children with ASD.

Based on an analysis of Ivar Lovaas's work, it is evident that the "applied behaviour analysis" approach has several drawbacks when it comes to addressing the challenges faced by autistic children. Consequently, we aimed to employ methodologies that encourage children with ASD to engage in "play" facilitated through technology. By utilizing technological means of interaction such as visual displays and physical robots, it becomes possible to elicit play and comfortable interactions from children with autism. One notable advantage is the sense of "safety" that arises from having the primary form of interaction with non-human entities.

2.2.3 Face emotions recognition

Face emotion recognition (FER) is a rapidly evolving field within the broader domain of artificial intelligence (AI) and computer vision. It involves the development of algorithms and systems capable of identifying and classifying human emotions based on facial expressions. The applications of FER are vast, ranging from enhancing human-computer interaction and virtual reality experiences to improving mental health diagnostics and surveillance systems.

The study of facial expressions and their correlation with emotions dates to the 19th century, with Charles Darwin's seminal work, "The Expression of the Emotions in Man and Animals" (1872). However, it wasn't until the 20th century that psychologist Paul Ekman laid the foundation for modern FER research by establishing the Facial Action Coding System (FACS) in 1978. FACS is a comprehensive taxonomy of facial muscle movements (action units) that correspond to specific emotions. This system has since served as the basis for developing computational models for FER.

Face emotion recognition techniques can be broadly categorized into two groups:

1. **Geometric-based methods**: These techniques focus on extracting facial landmarks (e.g., mouth corners, eyebrow positions) and computing geometric features, such as distances and angles between these landmarks. The extracted features are then used to train classifiers, such as SVM or k-NN, to recognize emotions.

2. **Appearance-based methods**: These techniques analyze the texture and appearance of the face by extracting features from the entire facial region or specific areas (e.g., eyes, mouth). Methods like Local Binary Patterns (LBP), Gabor filters, and Histogram of Oriented Gradients (HOG) have been employed for this purpose. More recently, deep learning techniques, particularly Convolutional Neural Networks (CNNs), have shown remarkable success in FER due to their ability to learn hierarchical features automatically.

Despite significant advancements, FER still faces several challenges, including:

- Variability in facial expressions: People express emotions differently, and subtle variations in expressions can be challenging to detect and classify accurately.
- Occlusion: Partially occluded faces, such as those covered by hair, glasses, or masks, can hinder the performance of FER systems.
- Illumination and pose variations: Changes in lighting conditions and head poses can affect the appearance of facial features, making emotion recognition more difficult.
- Cultural differences: Emotion expression and interpretation can vary across cultures, necessitating the development of culturally sensitive FER models.
- Real-time processing: Many applications require real-time FER, which demands efficient and computationally lightweight algorithms.

2.3 Literature Survey

2.3.1 Identification of autism spectrum disorder using deep learning and the ABIDE dataset.

This study[16] had two main objectives. Firstly, it aimed to categorize individuals with ASD and control participants according to their distinct functional connectivity neural patterns, as revealed through the analysis of rs-fMRI data. Secondly, the study sought to explore the neural patterns that were most associated with ASD classification. The findings were then examined in the context of the specific brain regions that distinguish ASD from control participants and previous research on ASD brain function.

The dataset includes 1112 individuals with autism and 1112 typically developing individuals.

The deep neural network achieved a mean classification accuracy of 70% (sensitivity 74%, specificity 63%) from cross-validation folds, and a range of accuracy of 66% to 71% in individual folds.

However, these differences in brain activity and connectivity are not consistent across all individuals with ASD and may be influenced by factors such as age, sex, and co-occurring conditions. Therefore, fMRI is not considered a reliable diagnostic tool for ASD on its own but may be used in combination with other diagnostic assessments and evaluations.

Overall fMRI research in ASD is still in its early stages and can't be reliable method for diagnosis.

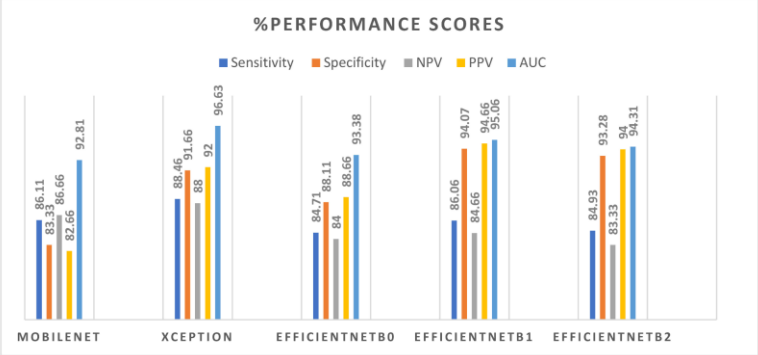
2.3.2 Identification of Autism in Children Using Static Facial Features and Deep Neural Networks

This method[17] proposes to detect autism by facial asymmetry in children by measuring the precise distance between pairs of facial landmarks and protrusions.

The dataset comprises a total of 3,014 facial images of children, evenly distributed across two categories: autistic (1,507 images) and non-autistic (1,507 images). The images are in jpg format and have a resolution of 224 × 224 × 3.

Five CNN-based binary ASD classifier models that use five pre-trained models, Mobile Net, Xception, EfficientNetB0, EfficientNetB1, and EfficientNetB2

They discovered that the Xception model outperformed the other models and achieved an AUC of 96.63%, a sensitivity of 88.46%, and an NPV of 88.00%.



Despite the promising results of this diagnosis method, it is not reliable method because the dataset contains inconsistent data:

- Highly photoshopped images
- Images of other syndromes like down syndrome and fragile x
- Varied ages and facial expressions

2.3.3 Autism Spectrum Disorder Detection with Machine Learning Methods

The objective of this study[18] is to categorize data related to ASD to offer a swift, accessible, and uncomplicated method for facilitating early diagnosis of ASD.

The study utilizes three distinct datasets for individuals with ASD, catering to children, adolescents, and adults. To classify the ASD data, three methods are employed: the KNN, SVM and the Random Forests.

To conduct the experiments, the data is randomly divided into training and test sets. The classification methods are tested using different random selections from the data, repeated 100 times.

The average values were used to evaluate the results. The findings indicate that SVM and Random Forests methods are effective in classifying autism spectrum disorder (ASD) data. Notably, the RF method achieved a 100% accuracy in classifying the data across all the mentioned datasets.

2.3.4 Theory of applied behavioural analysis and its application in teaching.

Fitzgerald et al. conducted a review[19] in 2008 which highlighted that the use of CBI led to academic improvements in students with mild or high incidence disabilities. These improvements were observed across various subjects such as reading, mathematics, writing, social studies, and science. CBI, by providing individualized instruction, immediate feedback, and interactive learning experiences, has shown promise as a beneficial tool for supporting students with disabilities in their academic progress. It's important to note that the effectiveness of CBI can vary depending on factors such as student characteristics, software quality, and instructional strategies employed. For the most accurate and up-to-date information

Multiple studies have focused on the creation of systems designed for individuals with ASD. For instance, Monibi and Hayes developed a library of virtual cards for activities tailored to autistic children, utilizing the Nokia N800 (Mocoto prototype) [20]. The virtual card library can be expanded by incorporating images or digital visuals, allowing for customization of activities based on various parameters such as card size, quantity, and audio cues.

In a study conducted by Hassan [21], a computer game was introduced to assist children between the ages of 9 and 14 who have ASD in comprehending the concept of money. The game was created using BYOB, an advanced variation of the game engine Scratch. The game was designed to be straightforward and user-friendly. The effectiveness of the game was assessed by testing it on a group of 4 girls and 5 boys with ASD. The results of the study demonstrated promising outcomes in terms of the children's ability to recognize the concept of money.

Aggarwal et al. [22] introduced a program called "LIFEis GAME" aimed at enhancing the communication skills of children with Autism. This program utilizes virtual characters, and players engage with an avatar to select the appropriate facial expression representing an emotion within a social scenario.

In a pioneering study, Yi Li et al. [23] introduced the idea of an adaptive game specifically designed for children with autism. They employed the Microsoft Kinect platform and Kinect games to collect information about the child's behaviour. By utilizing facial recognition technology, the game was able to adapt and respond to the child's emotional state, adjusting when the child displayed signs of sadness or happiness.

Applied Behaviour Analysis (ABA) is primarily based on the concept of reinforcing positive behaviour and actions with a reward system, while also avoiding the promotion of negative behaviour and unwanted actions by withholding any rewards.

As an approach to apply the ABA concept in CBI is the utilization of a Token Economy System [24], where the child is provided with a miniature BINGO board to earn tokens as they make progress during a session. Upon filling the board with tokens, the child can exchange them for a reinforcer such as watching a cartoon, playing with a toy, or having designated playtime. These rewarding techniques are extensively employed in ABA therapy with the ultimate objective of reinforcing desired behaviors in the child while gradually diminishing unwanted ones.

Alexis Bosseler and Dominic W. Massaro from the University of California [25] developed a similar application to the one introduced by Miller. This application, called Baldi, incorporates an animated face that presents vocabulary-related questions to children with autism. The child is then required to select an answer. The purpose of the animated face is to create a more human-like interaction between the application and the patient. In terms of feedback, the application utilizes happy, sad, and confused emoji faces as rewards, providing the patient with information about the outcome of their selection.

In general, research studies have shown positive effects of computer-based instruction on academic outcomes for students with disabilities, including improvements in reading, mathematics, writing, and other subject areas. These interventions can be tailored to the individual needs and abilities of students, allowing for personalized instruction and remediation.

2.3.5 Facial Expression Recognition Using Computer Vision

Facial Expression Recognition Based on Deep Learning: A Survey published in Applied Sciences in 2019[26]. The paper provides a comprehensive overview of deep learning techniques for facial expression recognition (FER), discussing various aspects in the field.

1. **Face detection**

Generally, face detection is the very beginning of an FER system. This technique is responsible for selecting the ROI of an input image that will be fed to the next steps of the FER system. Most reviewed papers used the classic Viola–Jones face detector [27] from 2004. Very few works used other face detectors, such as the Dlib library [28] and Multi-task Cascade Convolution Neural Network (MTCNN) [29].

The Viola–Jones face detector is a Machine Learning based approach where a cascade function is trained from a lot of positive images (images with faces) and negative images (images without faces). Haar features are used in this algorithm, and they are applied in all training images to find the best threshold which will classify the faces as positive or negative detections.

The Dlib face detector uses an ensemble of regression trees to regress the location of 68 facial landmarks from a sparse subset of intensity values extracted from an input image and, consequently, detect where the faces are.

As for MTCNN, it goes through three stages to output a proper face detection, and, in each stage, the face that is being analyzed goes through a CNN. The first stage obtains the candidate windows and their bounding box regression vectors, merging highly overlapped candidates [30]. The second stage feeds those candidates to another CNN, which rejects many false candidates. The third stage is similar to the second one, but it also outputs five facial landmarks' positions.

2. Network Architectures

The paper reviews several deep learning architectures that have been proposed for FER, including:

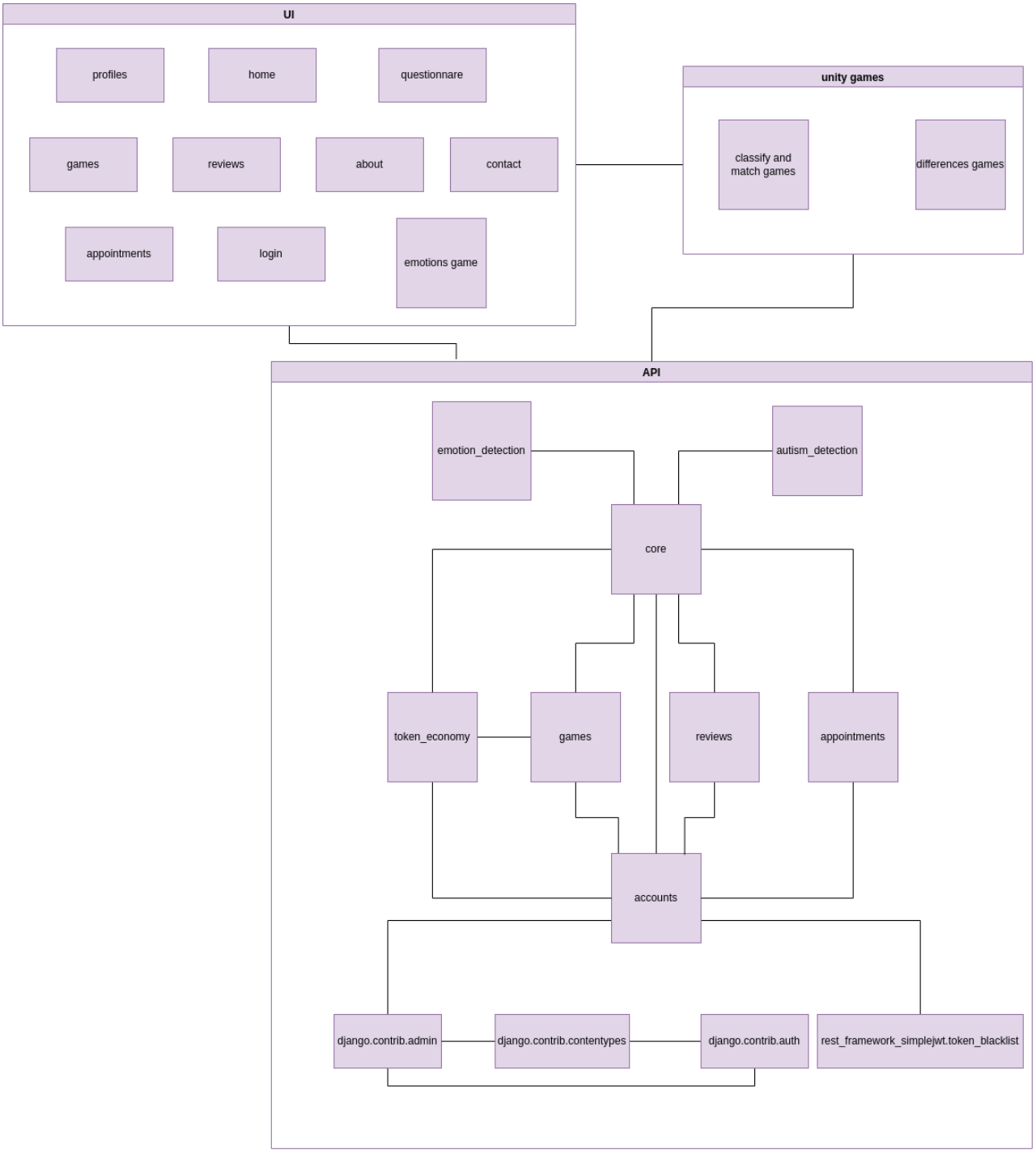
- Convolutional Neural Networks (CNNs): CNNs have become the dominant approach in FER due to their ability to learn hierarchical features automatically. The paper discusses various CNN architectures, such as AlexNet, VGGNet, and ResNet, which have been adapted for FER tasks.
- Recurrent Neural Networks (RNNs): RNNs, particularly Long Short-Term Memory (LSTM) networks, have been used for FER in video sequences, as they can model temporal dependencies between frames.
- Autoencoders: Autoencoders, such as Sparse Autoencoders (SAEs) and Variational Autoencoders (VAEs), have been employed for unsupervised feature learning in FER tasks.

Chapter 3

[The Proposed Solution]

This chapter presents our proposed solutions to the problems discussed in the preceding two chapters. We will delve into the intricacies of the system's design, providing a step-by-step account of the main features and their functions. Our aim is to offer a comprehensive overview of the design process, detailing each step taken to ensure the system's effectiveness and efficiency. The chapter will highlight the key elements of the system's design and how they were incorporated to address the identified challenges. With a focus on clarity and precision, we will provide an in-depth analysis of the system's design, offering a detailed understanding of its underlying architecture and functionality.

3.1 System overall



3.2 Autism screening model

3.2.1 Dataset

The dataset utilized in this study is derived from the Quantitative Checklist for Autism in Children (Q-CHAT) screening method. Specifically, the Q-CHAT-10, which comprises a set of ten questions, was employed. During the data collection process, the responses to these questions were mapped to binary values, which were subsequently assigned as class types.

Features mapping with Q-CHAT-10 screening method

Dataset variable	Description
A1	Child responding to you calling his/her name
A2	Ease of getting eye contact from child
A3	Child pointing to objects he/she wants
A4	Child pointing to draw your attention to his/her interests
A5	If the child shows pretense
A6	Ease of child to follow where you point/look
A7	If the child wants to comfort someone who is upset
A8	Child's first words
A9	If the child uses basic gestures
A10	If the child daydreams/stares at nothing

Autism Screening Child 2018 Version dataset (available on Kaggle) is a comprehensive collection of data related to autism screening in children. The dataset contains information on 509 children, including their age, gender, ethnicity, and scores on the Q-CHAT screening method. The dataset is in CSV format and contains 24 columns, including Child's Name, Age, Gender, Ethnicity, Jaundice, Family History of ASD, who completed the test, Q-CHAT-10 Score, and Class. The Class column is the target variable, with a value of "Yes" indicating the presence of potential ASD traits and "No" indicating the absence of ASD traits. This dataset is a valuable resource for researchers and healthcare professionals interested in autism screening and diagnosis in children.

3.2.2 Our proposed model

In this problem we need to a binary classification model to classify based on child data if he is autistic or not. So, we build an Artificial neural networks model as it used in many other applications and has almost good results.

The ANN architecture consists of four layers with varying numbers of neurons, activation functions, and other configurations. The model's hyperparameters include the loss function, optimizer, learning rate, number of epochs, and batch size.

3.2.2.1 Model Architecture

1. First Layer:
   - 256 dense neurons
   - He uniform initializer.
   - Rectified Linear Unit (ReLU) activation function.
2. Second Layer:
   - 128 dense neurons
   - ReLU activation function
3. Third Layer:
   - 64 dense neurons
   - ReLU activation function
   - 0.2 dropout rate
4. Output Layer:
   - 2 dense neurons
   - Softmax activation function

3.2.2.2 Hyperparameters

- **Loss function:** Cross-entropy
- **Optimizer:** Adam
- **Learning rate:** 0.0001
- **Number of epochs**: 25
- **Batch size:** 8

3.3 Face emotions Game

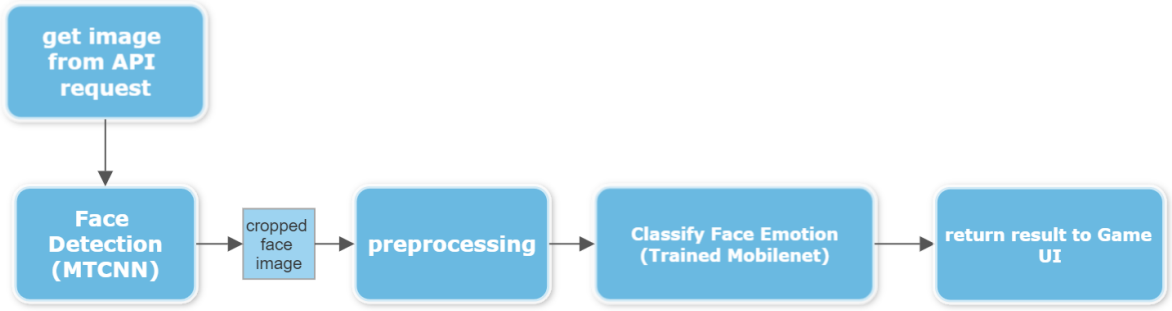
The game is designed to help children with autism spectrum disorder (ASD) improve their social communication skills. It does this by creating simulated real-world scenarios that allow children to express their emotions in a variety of contexts.

The goal is to provide a safe and supportive environment where children can practice and develop their social skills, which can be challenging for those with ASD. By engaging in the game's activities, children can learn how to communicate more effectively, understand social cues, and build stronger relationships with others.

Overall, the game aims to enhance the quality of life for children with ASD by improving their ability to interact with the world around them.

3.3.1 Base model

This face emotion recognition model is designed to detect and classify human emotions in images. The model is built using a two-step process: first, it employs the MTCNN (Multi-Task Cascaded Convolutional Networks) face detector to detect faces in the input image, and then it uses a transfer learning approach with the MobileNet model to classify the detected faces into different emotion categories. The dataset used for training and evaluation is the FER (Facial Expression Recognition) dataset.



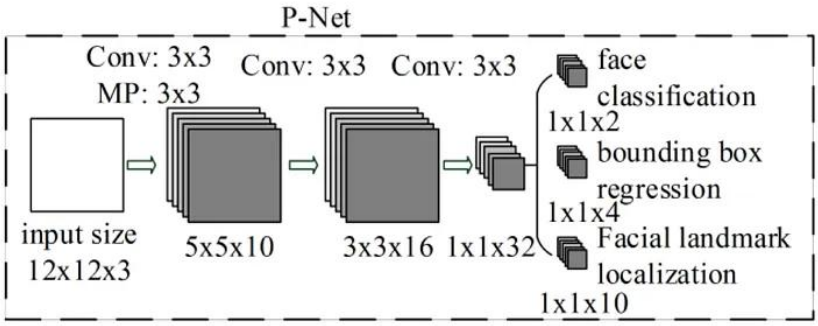
3.3.1.1 Face Detector [MTCNN]

MTCNN is a highly effective and efficient face detection algorithm that simultaneously detects faces and facial landmarks in an image. It consists of a three-stage cascaded network, where each stage refines the output of the previous stage.

3.3.1.1.1 MTCNN architecture

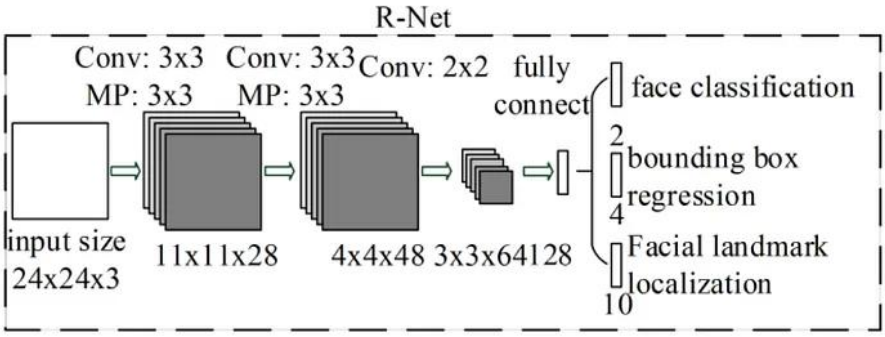
1. **Proposal Network (P-Net)**

The P-Net is the first stage of the MTCNN architecture. It is a fully convolutional network (FCN) that takes an input image and generates candidate bounding boxes for faces. The P-Net processes the input image at multiple scales to detect faces of different sizes. It uses a sliding window approach to scan the image and outputs a probability map indicating the presence of a face in each region. The P-Net also predicts bounding box regression offsets to adjust the candidate boxes.



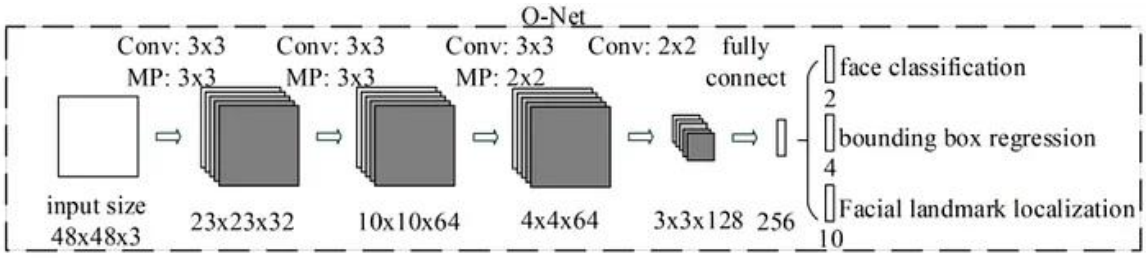
2. Refine Network (R-Net)

The R-Net is the second stage of the MTCNN architecture. It takes the candidate bounding boxes generated by the P-Net and refines them. The R-Net is a more complex convolutional neural network (CNN) compared to the P-Net. It processes each candidate bounding box and outputs a probability map and bounding box regression offsets, similar to the P-Net. The R-Net is responsible for rejecting false positives and refining the bounding boxes to improve their accuracy.



3. Output Network (O-Net)

The O-Net is the final stage of the MTCNN architecture. It further refines the bounding boxes and outputs facial landmarks. The O-Net is a more complex CNN than the R-Net and processes the refined bounding boxes from the previous stage. It outputs three types of information: a probability map indicating the presence of a face, bounding box regression offsets for further refinement, and facial landmark coordinates.



4. **Non-Maximum Suppression (NMS)**

   After each stage of the MTCNN, non-maximum suppression (NMS) is applied to reduce the number of overlapping bounding boxes. NMS works by selecting the bounding box with the highest probability and removing other boxes that have a high overlap (measured by the Intersection over Union, or IoU) with the selected box. This process is repeated until all boxes have been processed or removed.

3.3.1.1.2 Key Advantages of MTCNN

- **High detection accuracy:** MTCNN can detect faces with varying poses, scales, and occlusions, making it suitable for a wide range of applications.
- **Real-time performance:** MTCNN is computationally efficient, allowing for real-time face detection on devices with limited resources, such as mobile phones and embedded systems.
- **Joint face detection and facial landmark localization:** MTCNN not only detects faces but also outputs facial landmarks, which can be useful for tasks such as facial alignment, emotion recognition, and facial expression analysis.

3.3.1.2 Classifier Model [MobileNet]

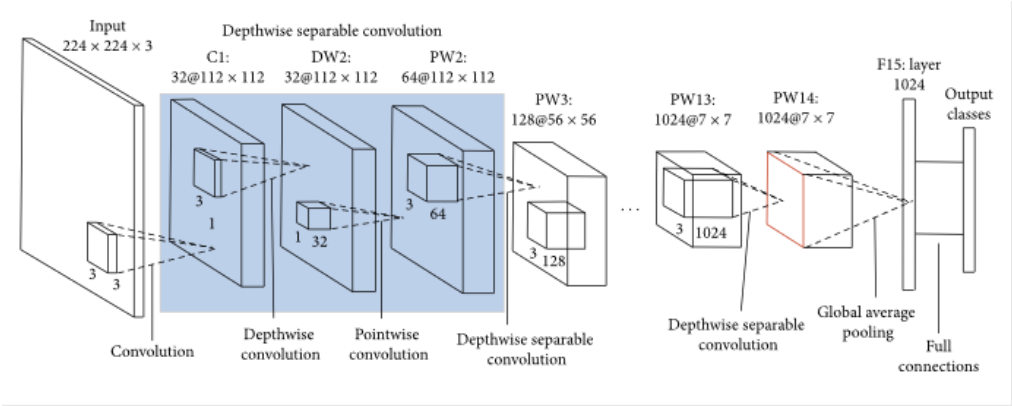
3.3.1.2.1 Dataset

The FER (Facial Expression Recognition) dataset is a large dataset of labelled facial expressions, consisting of 35,887 grayscale images of faces, each with a size of 48x48 pixels. The dataset is divided into seven emotion categories: anger, disgust, fear, happiness, sadness, surprise, and neutral.

We focus on the basic emotions in real life to improve child's social communication so, we used happiness, sadness, surprise, anger, and neutral classes only.

3.3.1.2.2 MobileNet model

MobileNet is a lightweight convolutional neural network architecture designed for mobile and embedded vision applications. It uses depth wise separable convolutions to reduce the number of parameters and computational cost compared to traditional convolutional layers. In this project, we use transfer learning to fine-tune a pre-trained MobileNet model on the FER dataset for emotion classification.



3.3.1.2.3 Fine tuning

By leveraging transfer learning with MobileNet, we can achieve state-of-the-art performance on our custom dataset with relatively less training time and computational resources compared to training a model from scratch.

We replaced the output layer to be 5 dense softmax output layer because our dataset contains 5 classes only.

We start training after config some hyperparameters:

- **Loss function:** Cross-entropy
- **Optimizer:** Adam
- **Learning rate:** 0.0001
- **Number of epochs**: 50
- **Batch size:** 32

3.3.2 Game UI

3.3.2.1 Game structure

The Emotion Classifier game is a React application designed to help individuals, particularly those on the autism spectrum, recognize and mimic different emotions. The game utilizes a machine learning model for facial emotion recognition to provide feedback and assess the player's performance.

3.3.2.2 User interface

The game's user interface consists of the following components:

3.3.2.2.1 Header

- **The header displays the current emotion to be mimicked, represented by an emoji and its corresponding adjective.**

3.3.2.2.2 Video Stream

- **The main section of the screen displays the video stream from the user's camera. The player can see themselves and their mimicking attempts in real-time.**

3.3.2.2.3 Progress Bar

- **A progress bar shows the remaining time to mimic the emotion. The game provides visual feedback by decreasing the progress bar's length over time.**

3.4 Pre academic skills Games

The second category of our system's games is specifically designed to enhance the skills of children with autism by providing targeted training in various concepts, such as identifying differences, recognizing shapes, classifying objects, and familiarizing them with essential tools in their surroundings. Each concept is divided into four progressive levels, gradually guiding the child through the learning process until they have thoroughly grasped the concept.

Our system comprises two distinct games: the first game emphasizes the concept of differences, while the second game focuses on fostering an understanding of shapes and objects through classification and matching exercises. These games have been meticulously crafted with visually appealing graphics, a token economy system, and a user-friendly interface to capture the child's interest. Additionally, audio elements have been incorporated to facilitate the learning process.

3.4.1 Differences Game

The main goal is to make sure that the child familiar with basic differences concepts like (big & small), (tall & short), (open & closed), etc. Our game design contains levels for (Big & small) and (tall & short) concepts each one has 4 levels.

3.4.2 Classify and Match Game

About this game designed to help children with autism develop their skills through engaging matching and classifying activities:

Introducing our innovative game, specifically designed to support the development of children with autism. We understand the importance of engaging and interactive activities for these children, and our game offers a unique combination of matching and classifying challenges that cater to their interests and abilities.

Our matching game features two distinct modes: colour and shape matching, which focuses primarily on colour recognition, and shape-only matching, which encourages children to identify and match objects based on their shape. This variety ensures that children can develop a wide range of skills while enjoying the game.

The classifying game is designed to help children differentiate between various categories, such as colours, animals, and transportation. With four levels of increasing complexity, children will be exposed to a diverse range of challenges that will keep them engaged and motivated to learn.

Each level in the matching game introduces a greater number of objects, ensuring that children continue to learn and grow as they progress through the game. In the classifying game, every level offers a unique experience, keeping the gameplay fresh and exciting.

3.5 Functional / Non-functional Requirements

3.5.1 Functional Requirements

3.5.1.1 User Requirements

3.5.1.1.1 The user shall be able to register to our system.

3.5.1.1.2 The user shall be able to login.

3.5.1.1.3 The user shall be able to view his profile.

3.5.1.1.4 The user shall be able to edit his profile.

3.5.1.1.5 The user shall be able to delete his profile.

3.5.1.1.6 The user shall be able to take a test to check if his child has autism or not.

3.5.1.1.7 The user shall be able to see the nearest specialist.

3.5.1.2 Child Requirements

3.5.1.2.1 The child shall be able to play games Under the supervision of specialist for skills development.

3.5.1.2.2 The child shall be able to view his token economy that the system adds to him.

3.5.1.2.3 The child shall be able to view games and his profile.

3.5.1.3 Specialist Requirements

3.5.1.3.1 The specialist shall be able to view his profile and edit on it.

3.5.1.3.2 The specialist shall be able to check his appointments.

3.5.1.3.3 The specialist shall be able to process his appointments by accepting or decline.

3.5.1.3.4 The specialist shall be able to edit his timetable.

3.5.1.4 Parent Requirements

3.5.1.4.1 The parent shall be able to view his profile and edit it.

3.5.1.4.2 The parent shall be able to control his child's profile by editing or deleting it.

3.5.1.4.3 The parent shall be able to add a new child to the system (up to 10 children).

3.5.1.4.4 The parent shall be able to book or cancel an appointment.

3.5.1.4.5 The parent shall be able to check if his appointment accepted or not.

3.5.1.4.6 The parent shall be able to review his specialist.

3.5.1.4.7 The parent shall be able to edit his review.

3.5.1.4.8 The parent shall be able to delete his review.

3.5.2 Non-Functional Requirements

3.5.2.1 Implementation Requirements:

In implementing the whole system, we will use React JS as the front-end technology, with Django as the server-side scripting language for database connectivity. The backend will be developed using PostgreSQL as the database management system. Additionally, Pandas, NumPy, Keras and Open-CV will be used for developing machine learning models, and Unity will be used for developing games.

3.5.2.2 Efficiency Requirement:

When Autism Care System will be implemented, the users will detect and treat the autism by answering many questions and the system will send these answers to machine learning model that can detect autism and treating stage by booking appointments with specialist and the specialist can follow the child by many ways such as his score in games, the system should be able to upgrade without disturbance to the service.

3.5.2.3 Reliability Requirement:

The system should accurately perform registration, user validation and search, shall be compatible with any hardware, should perform calculations and provide feedback quickly, user can contact the staff to feed them back with any complaints.

3.5.2.4 Usability Requirement:

The system is designed for a user - friendly environment so that users can perform the various tasks easily and in an effective way. shall be able minimize the rate of errors generated by users, should be developed to be simple and efficient for the end users and should be easy to understand.

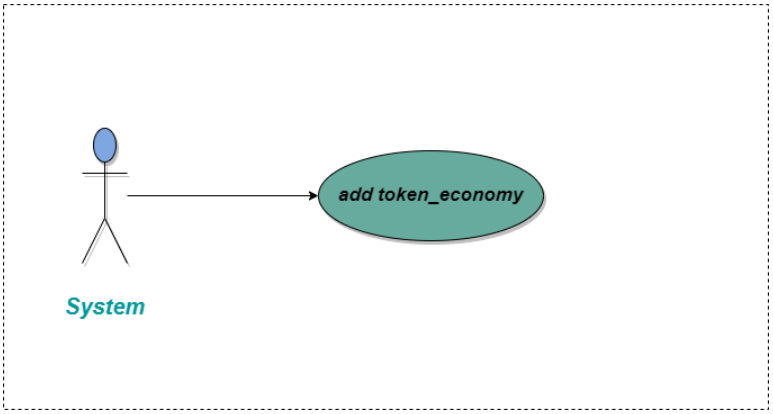
3.6 System Analysis & Design

3.6.1 Use cases.

A use case is a description of a specific way in which a system can be used to achieve a particular goal or objective. It is a technique used in software engineering and other fields to capture the functional requirements of a system or product from the perspective of stakeholder.

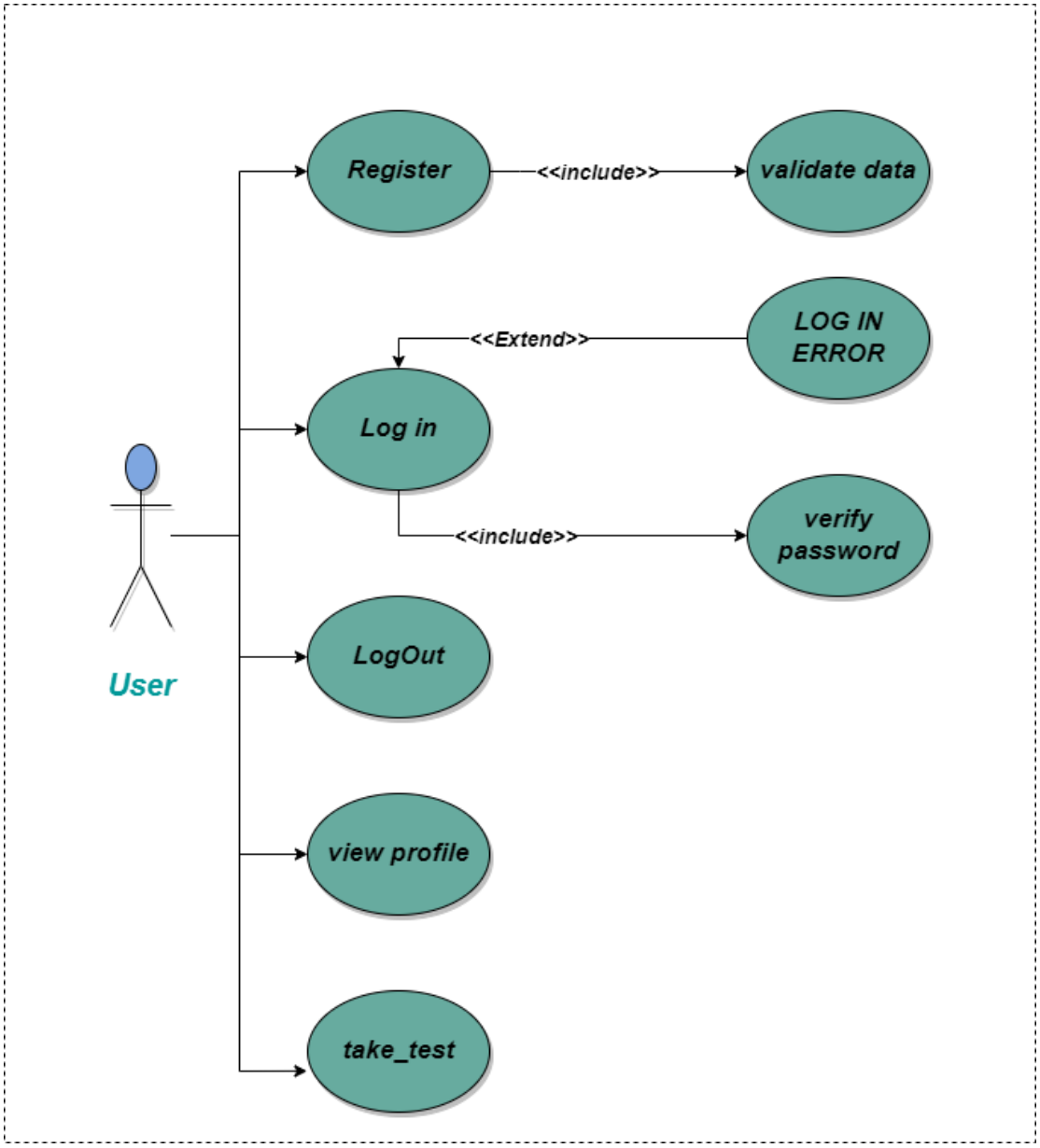
In a use case, the focus is on the interactions between the user or actor and the system. It typically includes a description of the preconditions, steps, and postconditions that are necessary for the user to achieve their goal using the system or product. Use cases can be represented in various ways, such as in natural language, diagrams, or formal notation.

3.6.1.1 System use cases.

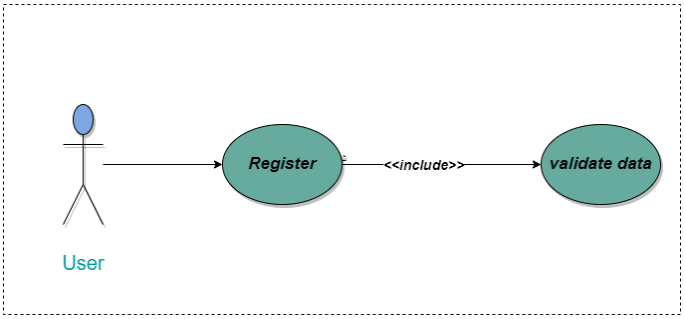


Actor	System
Description	System can add a gift to the autism_child if he wins a game
Precondition	Child must win a game
Postcondition	New Token_Economy in a child profile
Main successful scenario	1. A child wins a new game. 2. A system select token_economy. 3. A system adds this token_economy. 4. New token_economy in the child profile
unsuccessful scenario	None.

3.6.1.2 User use cases.

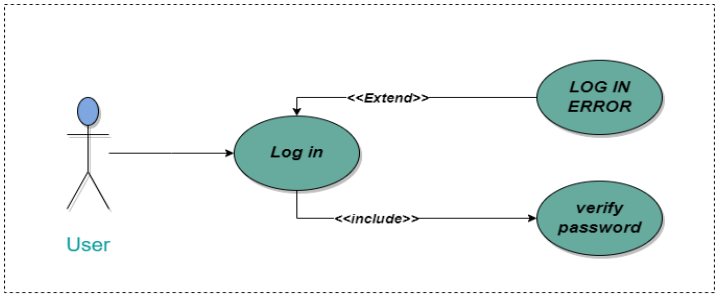


3.6.1.2.1 Registration



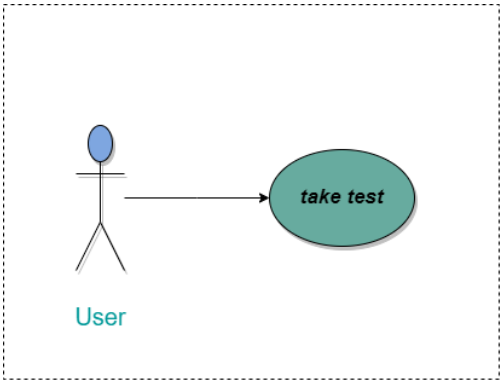
Actor	User
Description	New users create an account can use more options in our website
Precondition	none
Postcondition	User has an account on our system
Main successful scenario	1. User enters signup page. 2. He fills fields with a correct data . 3. Now he has a new account .
unsuccessful scenario	The user may fill fields with wrong data and cannot use our system

3.6.1.2.2 Login



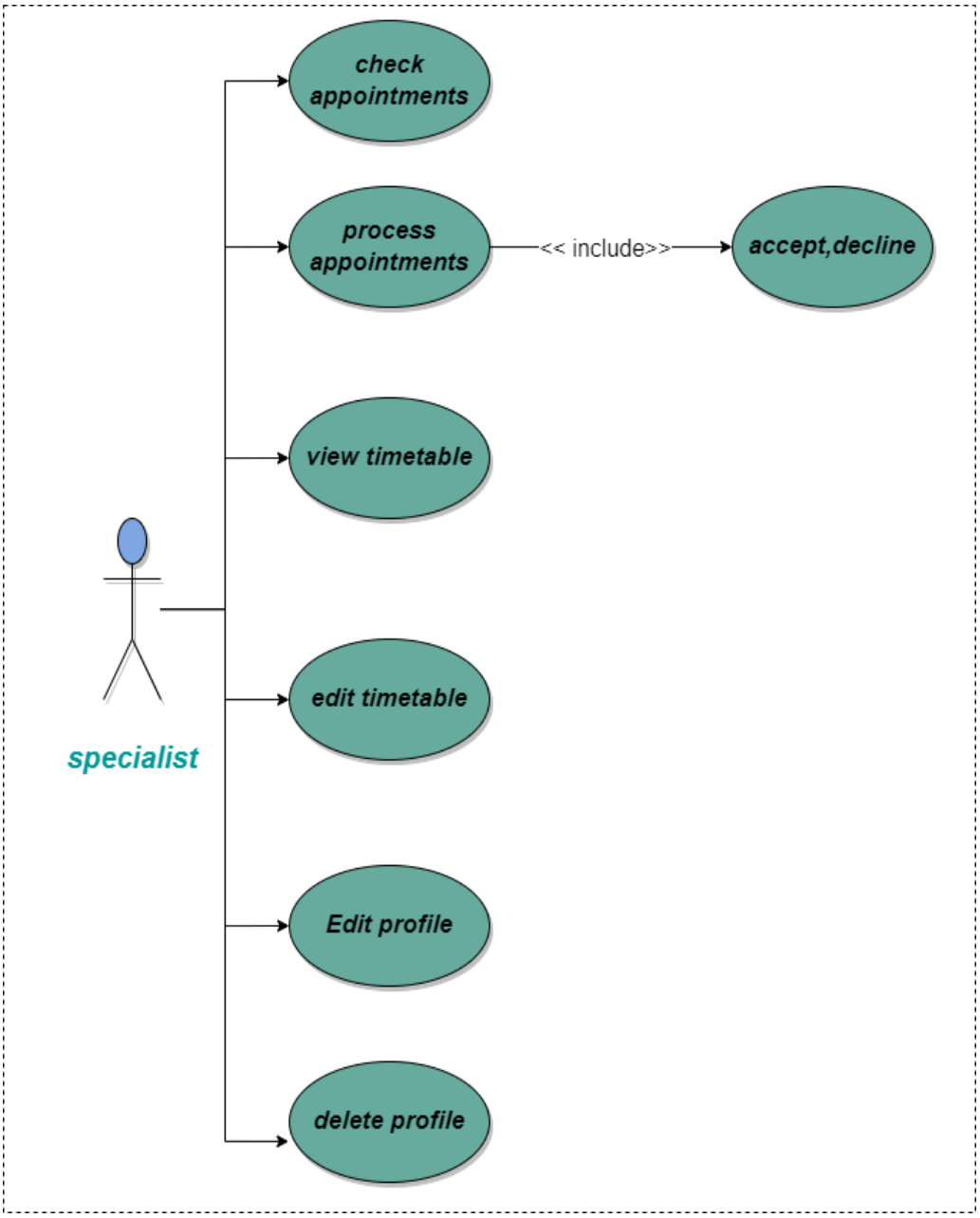
Actor	User
Description	An existing user want to login to our system
Precondition	The user must register
Postcondition	User can use our system
Main successful scenario	1. User enters log in page. 2. He puts his correct username and password. 3. The system will verify password and username and if true He enters our system and can use it.
unsuccessful scenario	If user does not have an account or puts wrong data, he cannot use the system

3.6.1.2.3 Take test.

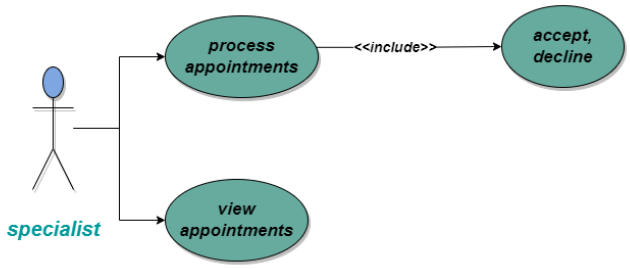


Actor	User
Description	The user can answer the questions in the questionnaire page to check if his child autistic patient or not .
Precondition	The user has an account
Postcondition	The system will predict and show the result to the user
Main successful scenario	1. User log in our system . 2. He answers the questions with a specific answer . 3. The system will know him if his child has autism or not . 4. If his child has an autism , the system will see him many ways to develops his skills and specialists to help him.
unsuccessful scenario	none

3.6.1.3 Specialist use cases.

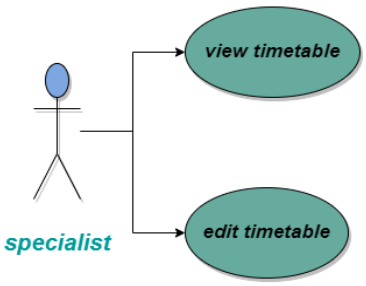


3.6.1.3.1 View and process appointment.



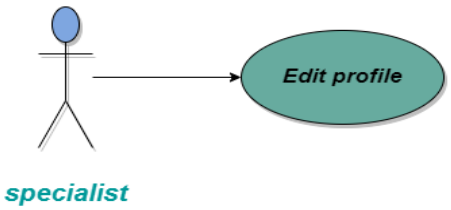
Actor	Specialist
Description	The specialist can check his reservations he has and take a decision of approval or rejection.
Precondition	He must sign up as a specialist.
Postcondition	Any reservation has been answered with approval or rejection
Main successful scenario	1. The specialist enters appointments page and view his appointments. 2. If he has appointments , he must take a decision with accept or decline
unsuccessful scenario	none

3.6.1.3.2 View and edit timetables.



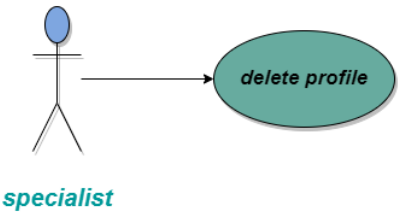
Actor	**Specialist**
Description	The specialist can view his timetable and edit on it .
Precondition	He must sign up as a specialist.
Postcondition	New timetable to the specialist
Main successful scenario	1. The specialist enters appointments page. 2. The system returns timetable of this specialist. 3. He writes a new data in the timetable. 4. The system will save the new timetable.
unsuccessful scenario	None

3.6.1.3.3 Edit profile.



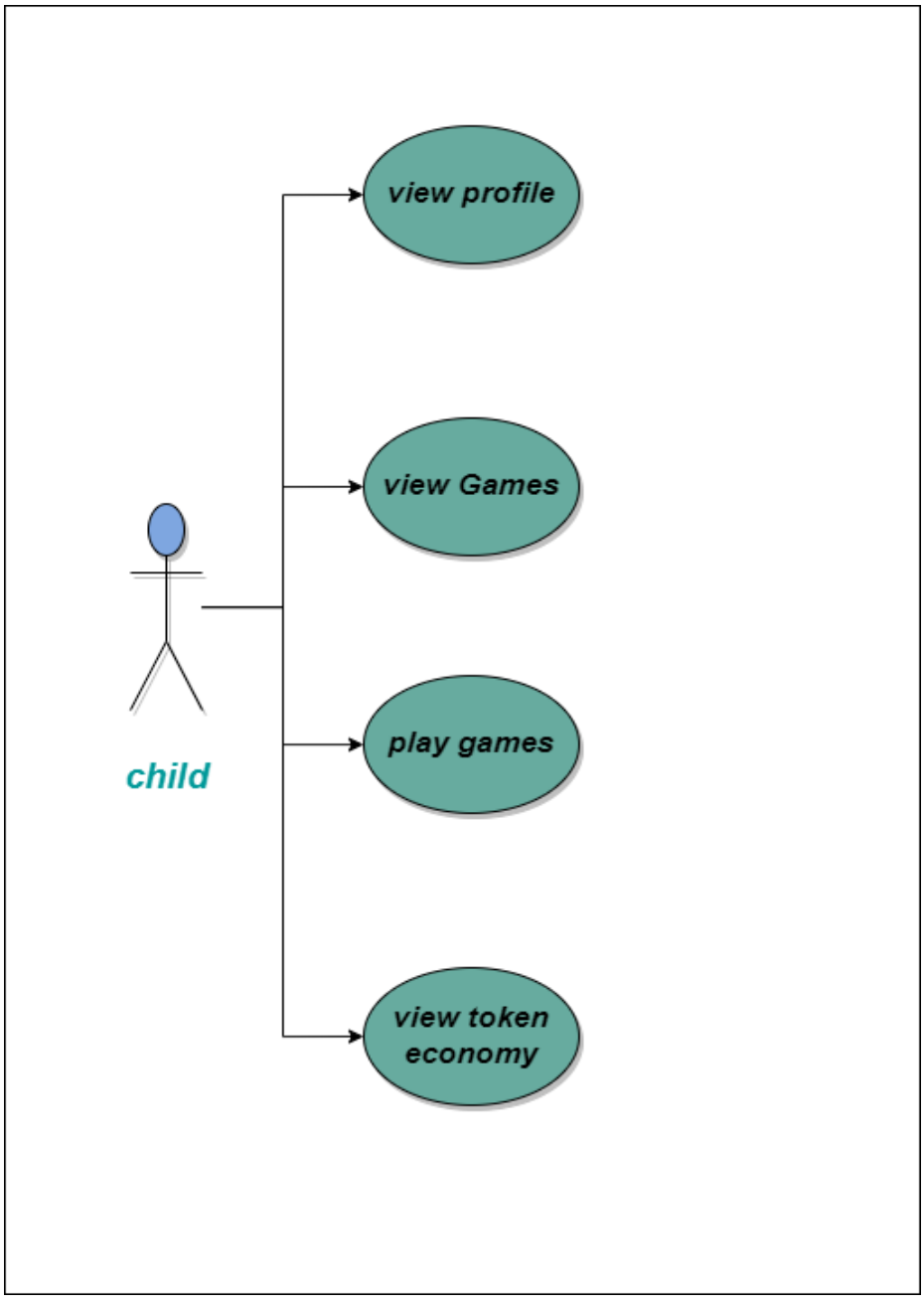
Actor	Specialist
Description	The specialist can edit his profile .
Precondition	He must sign up as a specialist.
Postcondition	New data to this doctor in his profile
Main successful scenario	1. The specialist enters profiles page. 2. Choose edit profile and write new data. 3. New data about this specialist in his profile.
unsuccessful scenario	The specialist may write wrong data about him, and the system refuse editing.

3.6.1.3.4 Delete profile.

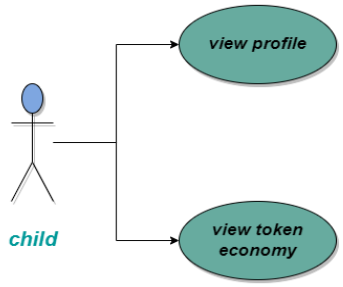


Actor	Specialist
Description	The specialist can delete his profile .
Precondition	He must sign up as a specialist.
Postcondition	No profile to this specialist
Main successful scenario	1. The specialist enters profiles page. 2. Choose delete profile. 3. The profile deleted successfully.
unsuccessful scenario	None

3.6.1.4 Child use cases.

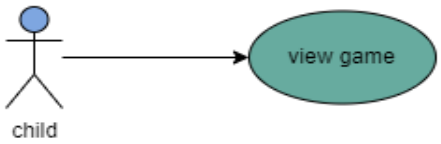


3.6.1.4.1 View profile and token economy.



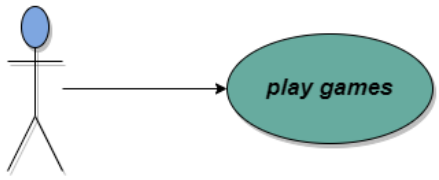
Actor	Child
Description	The child can see his or her profile and token economy that the system adds it to them.
Precondition	He must sign up as a child
Postcondition	The system returns the profile and token economy page
Main successful scenario	1. The child or his parent enters profiles page. 2. He can view his profile or choose token economy to 3. See it .
unsuccessful scenario	None

3.6.1.4.2 View game.



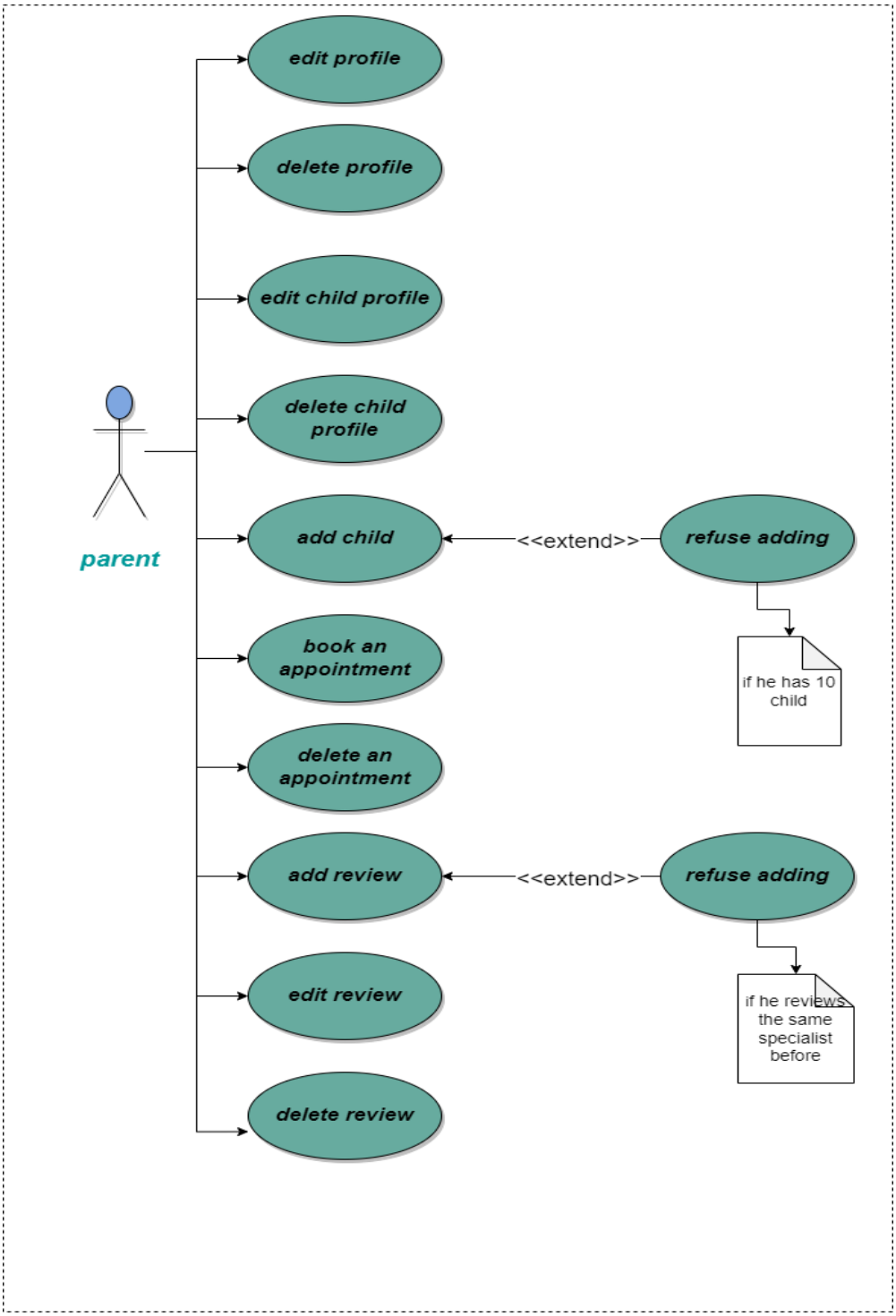
Actor	Child
Description	The child can view games to choose which game will he play.
Precondition	He must sign up as a child
Postcondition	The system returns the games
Main successful scenario	1. The child enters treatment page. 2. The system will return games. 3. Child can view games and its levels .
unsuccessful scenario	None

3.6.1.4.3 Play game.

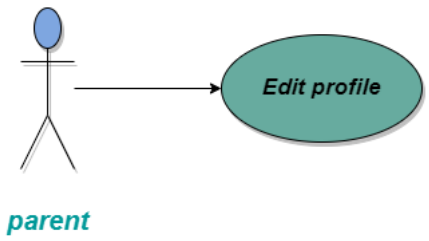


Actor	Child
Description	The child can play games to treatment
Precondition	He must sign up as a child
Postcondition	The child plays new games and if he wins the system adds new token economy.
Main successful scenario	1. The child enters treatment page. 2. He chooses the game which the specialist selects to him. 3. He plays can play one level or more than level if he can win it quickly. 4. The systems add new token economy to him.
unsuccessful scenario	None

3.6.1.5 Parent use cases.

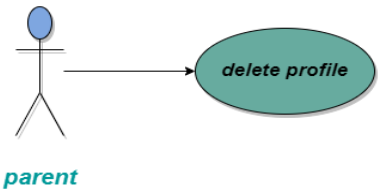


3.6.1.5.1 Edit profile.



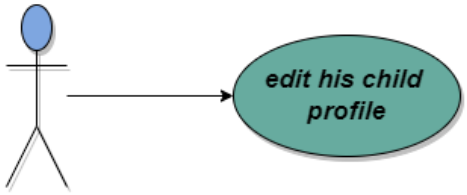
Actor	Parent
Description	The parent can edit his profile .
Precondition	He must sign up as a parent.
Postcondition	New data to this parent in his profile
Main successful scenario	1. The parent enters profiles page. 2. Choose edit profile and write new data. 3. New data about this parent in his profile.
unsuccessful scenario	The parent may write wrong data about him, and the system refuse editing.

3.6.1.5.2 Delete profile.



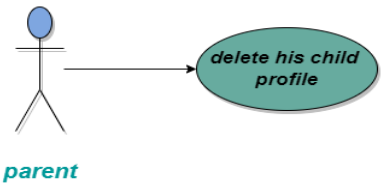
Actor	parent
Description	The parent can delete his profile .
Precondition	He must sign up as a parent.
Postcondition	No profile to this parent
Main successful scenario	1. The parent enters profiles page. 2. Choose to delete profile. 3. The profile deleted successfully
unsuccessful scenario	None

3.6.1.5.3 Edit child profile.



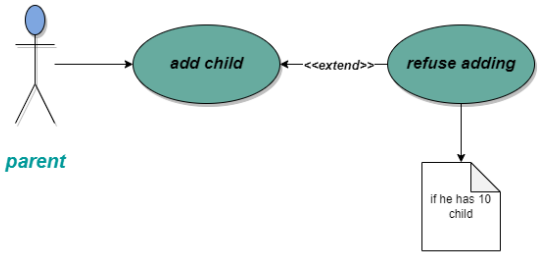
Actor	parent
Description	The parent can edit his child profile .
Precondition	He must sign up as a parent.
Postcondition	New data to the child in his profile
Main successful scenario	1. The parent enters profiles page. 2. Choose his child profile. 3. choose edit profile and write new data. 4. New data about this parent in his profile.
unsuccessful scenario	The parent may write wrong data about child and the system refuse editing.

3.6.1.5.4 Delete child profile.



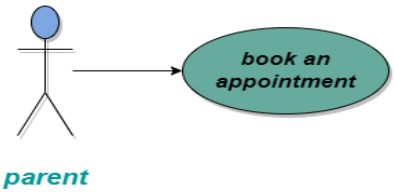
Actor	parent
Description	The parent can delete his child profile .
Precondition	He must sign up as a parent.
Postcondition	No profile to his child
Main successful scenario	1. The parent enters profiles page. 2. Choose his child profile. 3. Choose delete profile. 4. The profile deleted successfully
unsuccessful scenario	None

3.6.1.5.5 Add child.



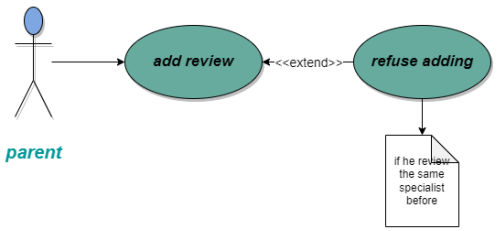
Actor	parent
Description	The parent can add new child to the system if he doesn't have more than 10 children.
Precondition	He must sign up as a parent.
Postcondition	New child will be added to the system
Main successful scenario	1. The parent enters add child page 2. Write his child data. 3. New child added to the system
unsuccessful scenario	The parent may write wrong data about child or if he has 10 child and the system refuse adding.

3.6.1.5.6 Book an appointment.



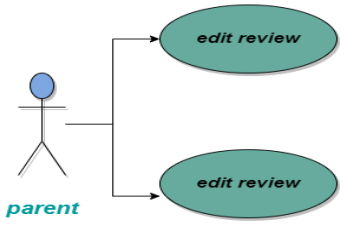
Actor	parent
Description	The parent can book an appointment with a specialist to show his child to him.
Precondition	He must sign up as a parent and has at least one child.
Postcondition	The parent booked an appointment
Main successful scenario	1. The parent enters appointment page. 2. Open the timetable of the specialist. 3. Choose available appointment. 4. An appointment has booked to the parent
unsuccessful scenario	The parent chooses unavailable appointment and the specialist decline it

3.6.1.5.7 Add review.



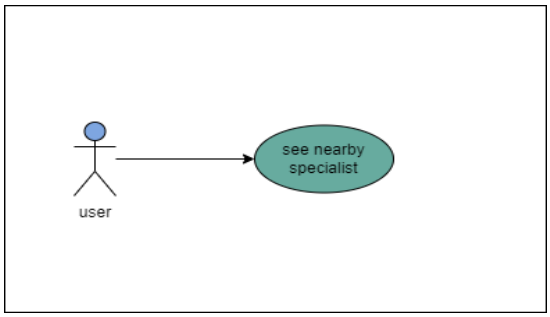
Actor	parent
Description	The parent can add review to any specialists one time .
Precondition	He must sign up as a parent.
Postcondition	New review will be added to the specialist
Main successful scenario	1. The parent enters appointment page. 2. Choose the specialist which he wants to review. 3. Write a new review to the specialist
unsuccessful scenario	The parent wants to add a review to specialist who reviewed before by the same parent

3.6.1.5.8 Edit or delete review.



Actor	parent
Description	The parent can delete or edit his review about the specialist.
Precondition	Parent added review
Postcondition	Deleted or edit review successfully
Main successful scenario	1. The parent enters appointments page. 2. Choose the specialist and choose to edit or delete. 3. If delete the system will delete it 4. If edit the parent will write the new review and it will be updated
unsuccessful scenario	None

3.6.1.5.9 Nearby specialist.



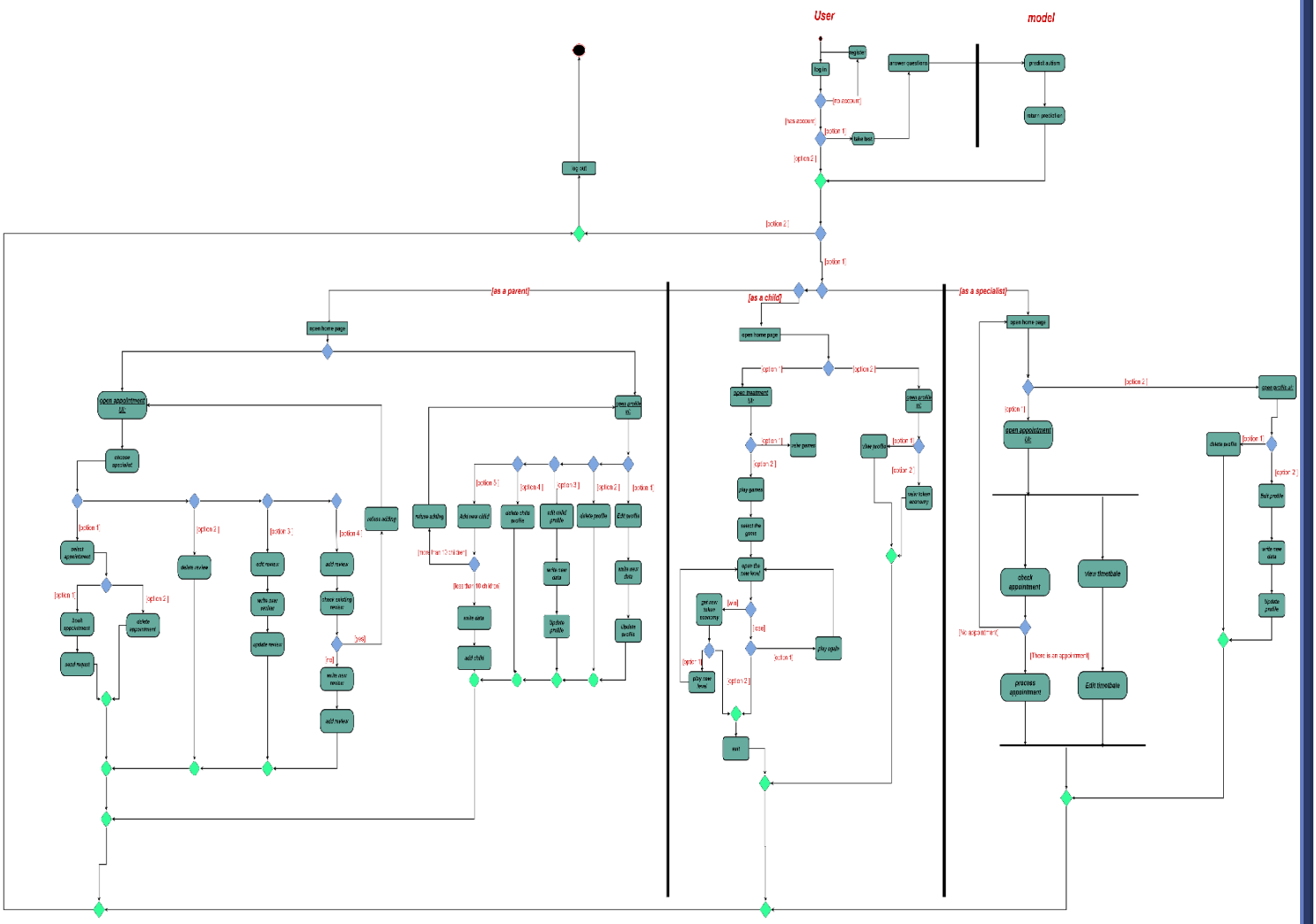
Actor	user
Description	The user can see or ask the system about the nearest specialist
Precondition	log in to the system, take a test and the result is positive
Postcondition	the system will return the nearest specialist
Main successful scenario	1. users log in to the system. 2. users take a test, and the result must be positive. 3. user open appointment page. 4. the system will find the city of the user. 5. the system will return the specialist around the user
unsuccessful scenario	the system may fail in find the user's location and return specialist far from user

3.6.2 Activity diagrams.

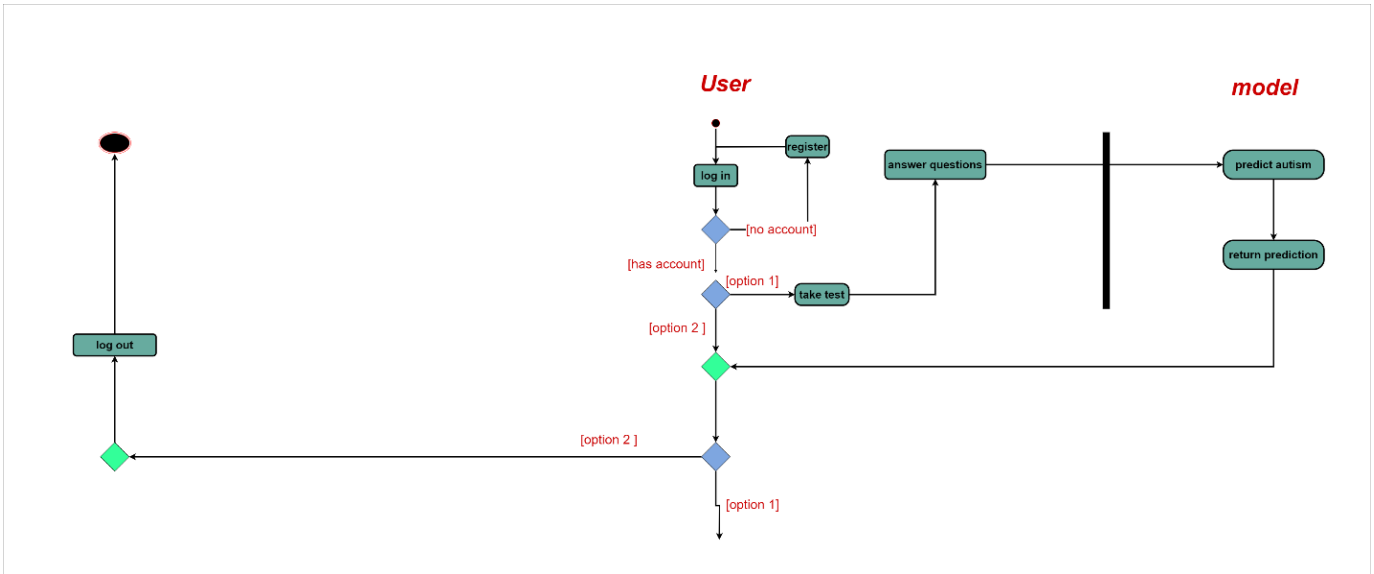
An activity diagram is a type of behaviour diagram in the Unified Modelling Language (UML) that shows the flow of activities or actions in a system, process, or workflow. It is used to model business processes, software applications, and other complex systems where there are multiple activities or actions that need to be coordinated.

We use it to show steps or activities to user to do it to do specific function , we also describe the steps to achieve any functions in use case diagram .

We also use the successful scenario in use case.



3.6.2.1 User activity diagram.



❖ In first user can login to the system by:

1. put username and password.
2. The system will validate username and password if correct data he will log in successfully.

❖ If he doesn't have an account, he must register at first by:

1. Choose his type [specialist, parent, or child]
2. Put his correct registration data.
3. The system will validate data.
4. The system will add user to data base.
5. Now the user can login to our system

❖ After entering to the system, the user has two options:

A. Take a test to check his child is belong to autism child or not by:

1. Open test page and answer all questions by specific answers .
2. The machine learning model will take answers and return a result .

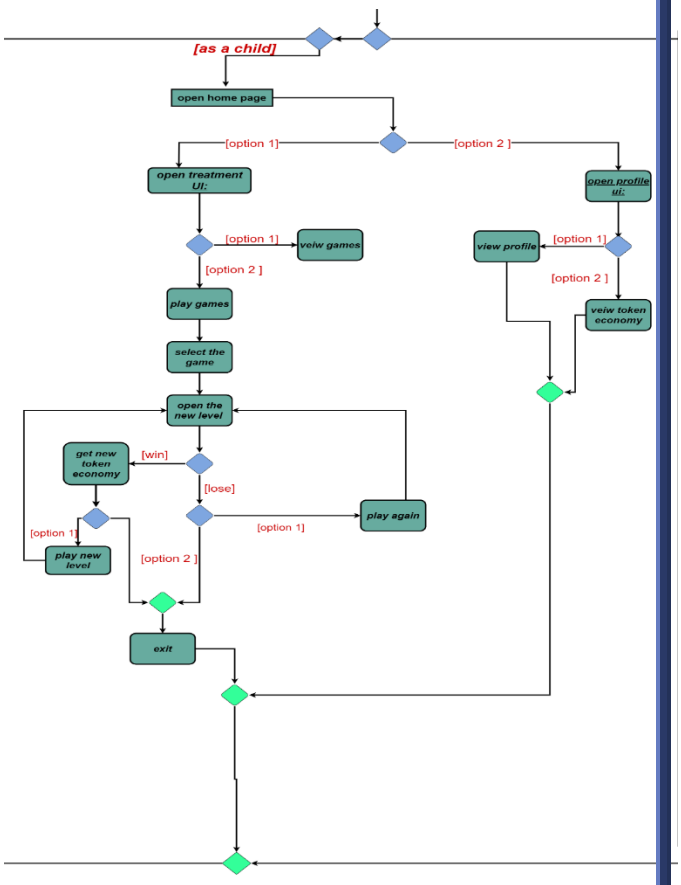
B. The second option is continued in system as:

1. Child

➢ **In the profile page:**

The child can view his profile by opening profile page and the system will return his data.

Also, can see his token economy in his profile page .



➢ **In treatment page:**

He can view games and play games by:

1. Select game.
2. Choose the level.
3. Play and If win ,the system will add new token economy and save last level.
4. If lose, he can play again or exit the game.

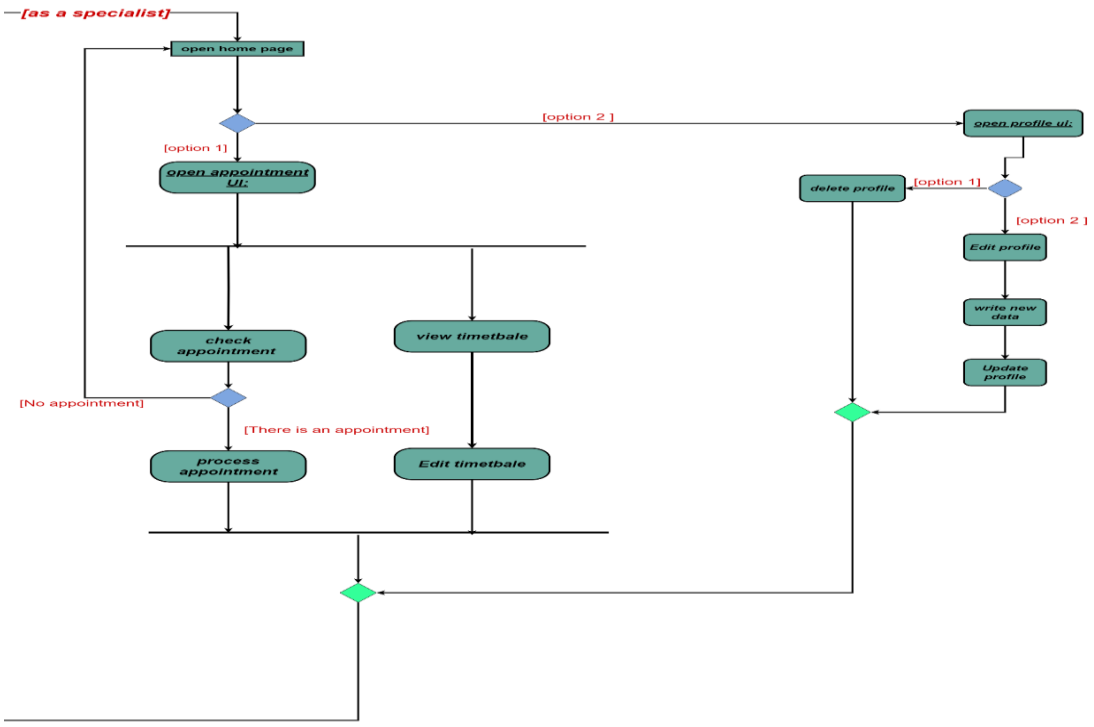
2. Specialist

➢ **Any specialist can edit or delete his profile by:**

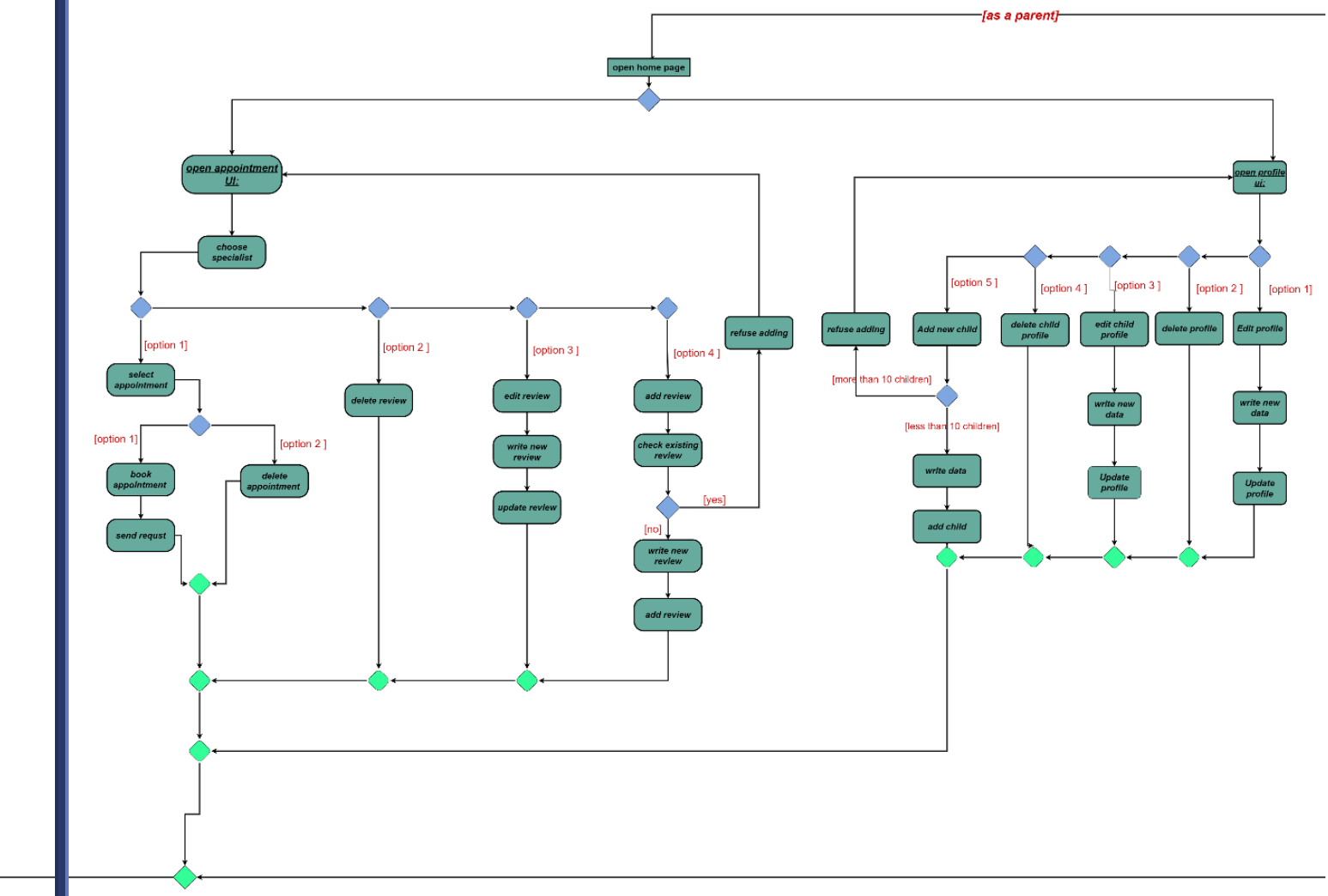
1. Open profile page .
2. Choose what he needs if edit:
   2.1. The system will return edit page with his old data.
   2.2. He must edit his data.
   2.3. The system will update his data in database and return success message.
3. If delete, he chooses delete profile and the system will delete it from database successfully.

➢ **Also, he can view appointment and process it by :**

1. Open appointment page .
2. If there is new appointment he can process if by accepting or declining appointment
3. If not, he can view his timetable and edit in it if he wants by:
   3.1. Choose the time that he wants to edit.
   3.2. Write new time.
   3.3. The system will update the time and send success message.



3. Parent


[as a parent]
open home page
open appointment UI:
choose specialist
[option 1]
select appointment
[option 1]
book appointment
send requst
[option 2]
delete appointment
[option 2]
delete review
[option 3]
edit review
write new review
update review
[option 4]
add review
check existing review
[yes]
[no]
write new review
add review
refuse adding
open profile ui:
[option 5]
refuse adding
[more than 10 children]
[less than 10 children]
Add new child
write data
add child
[option 4]
delete child profile
[option 3]
edit child profile
write new data
Update profile
[option 2]
delete profile
[option 1]
Edit profile
write new data
Update profile

- **The parent has the most functions in our system because he does many things such as:**

1. Edit or delete his profile by :
    a. Open profile page .
    b. Choose the transaction editing or deleting.
    c. If edit he must write new data and the system will update his data in database
    d. If delete the system will delete this user .

2. Edit or delete his child profile by:
    a. Open profile page.
    b. Select his child.
    c. Choose the transaction editing or deleting.
    d. If edit he must write new data and the system will update his data in database
    e. If delete the system will delete this user.

3. Add new child by :
    a. Open profile photo and choose add child.
    b. Put new child data.
    c. The system will check the num of children belong to this parent if more than 10 the adding will refuse else the system will accept adding.
    d. The system will add new child to database and return success message.

4. Book an appointment by :
    a. Open appointment page
    b. Select The appropriate date and book it.
    c. The system will send the request to the specialist, and he will process it.

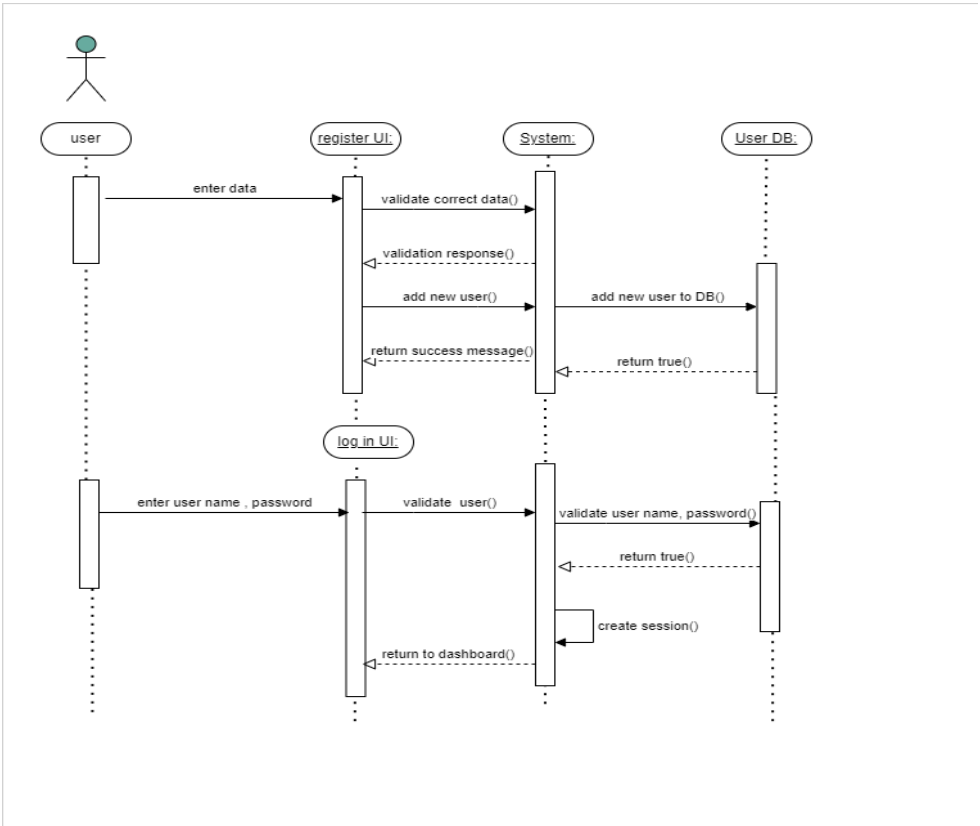
5. Add review to specialist :
    a. Select the specialist.
    b. Write his data and put his review.
    c. The system will check if this parent reviews this specialist before if yes, he cannot review again if not the review will be added successfully.

6. Edit or delete review:
    a. Select the specialist who he reviews before
    b. Choose the transaction [edit review , delete review]
    c. If edit he will put new review and the system will update it
    d. If delete the system will delete it successfully

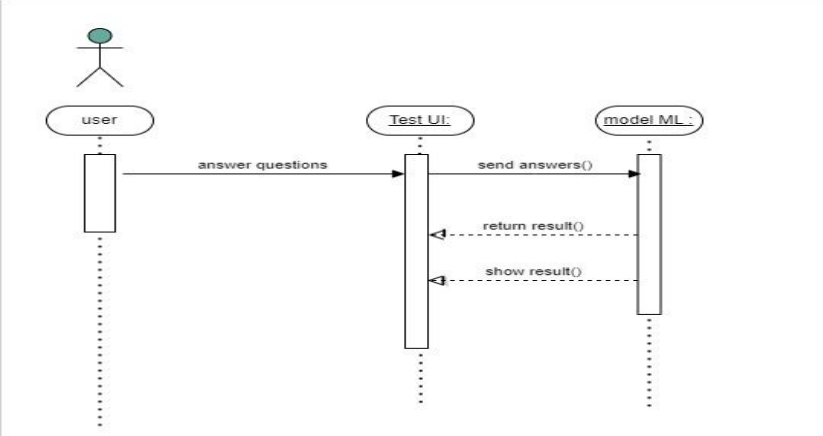
3.6.3 Sequence diagrams

Sequence diagrams illustrate the interactions between objects or components in a system in a time-ordered manner. They are used to show the flow of messages and method calls between objects, along with the order in which they occur. Sequence diagrams are often used to model use cases or scenarios and can be used to identify potential issues or errors in the interaction between system components.

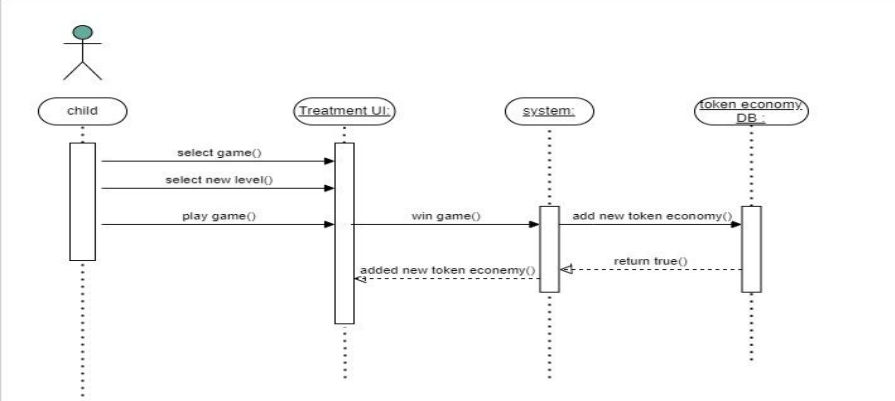
3.6.3.1 Registration and login.



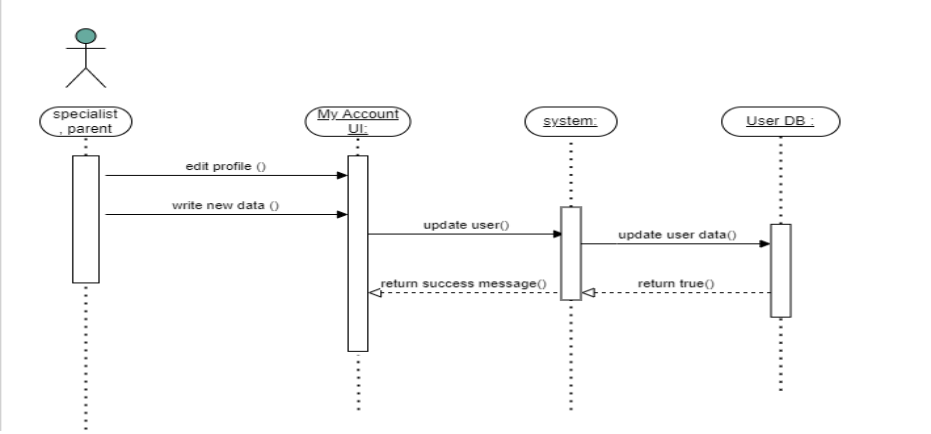
3.6.3.2 Take test.



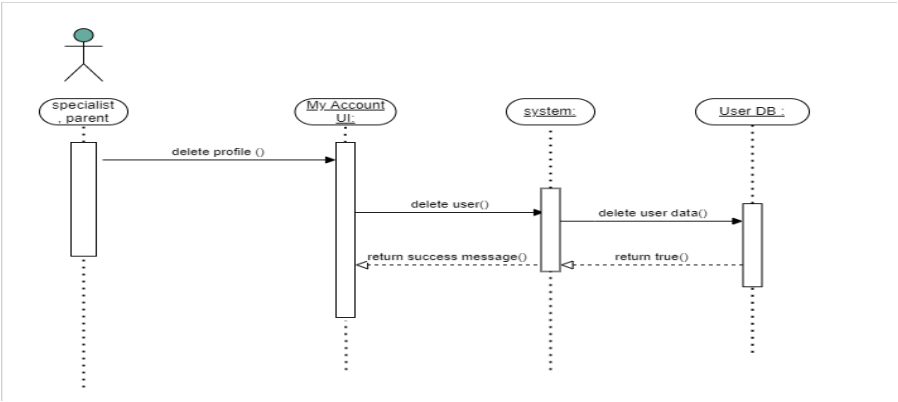
3.6.3.3 Play game.



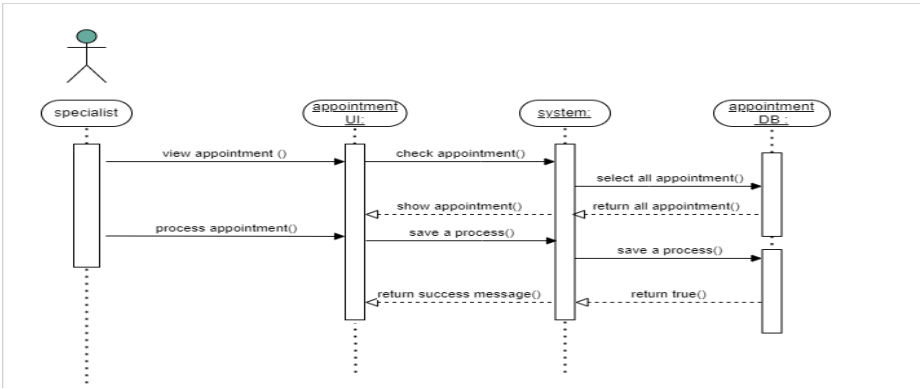
3.6.3.4 Edit profile.



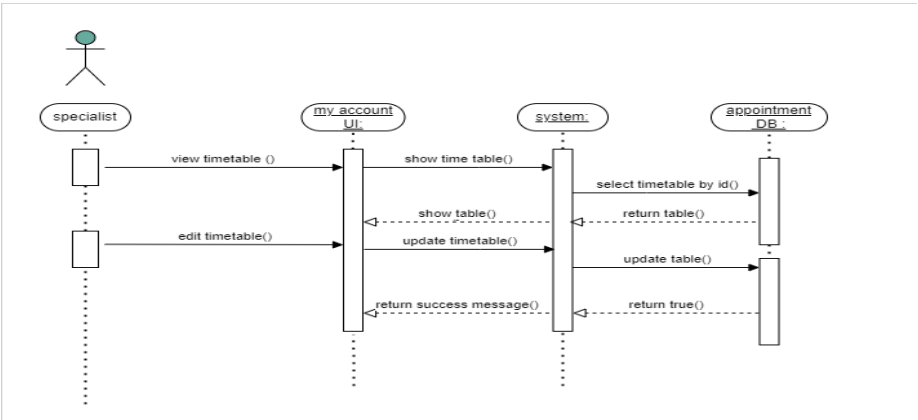
3.6.3.5 Delete profile.



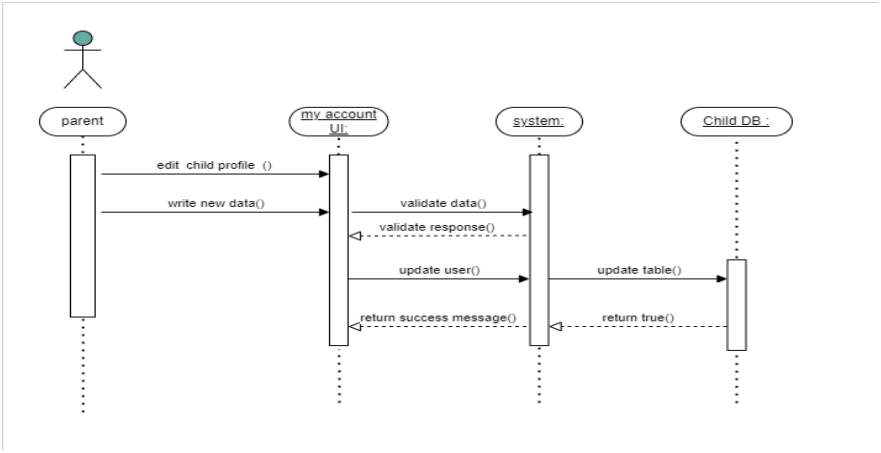
3.6.3.6 View and process appointment.



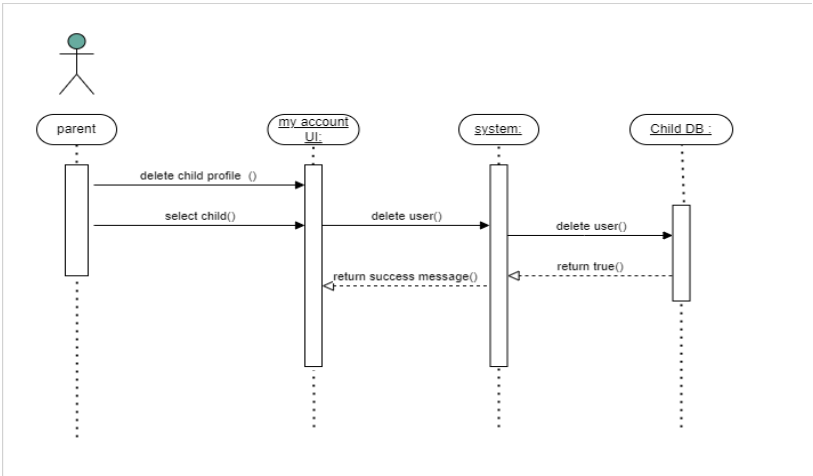
3.6.3.7 View and edit timetable.



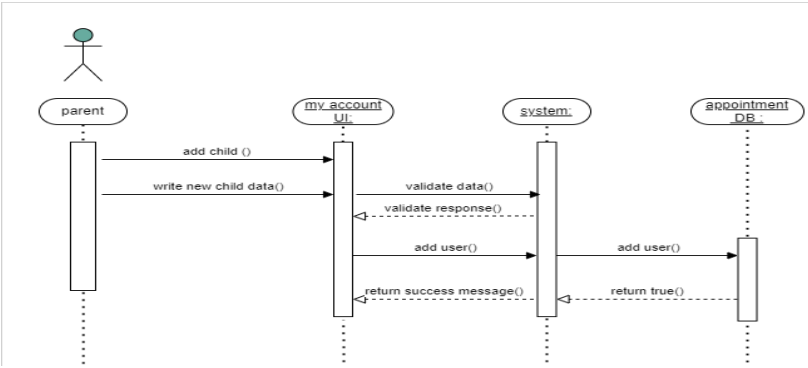
3.6.3.8 Edit child profile.



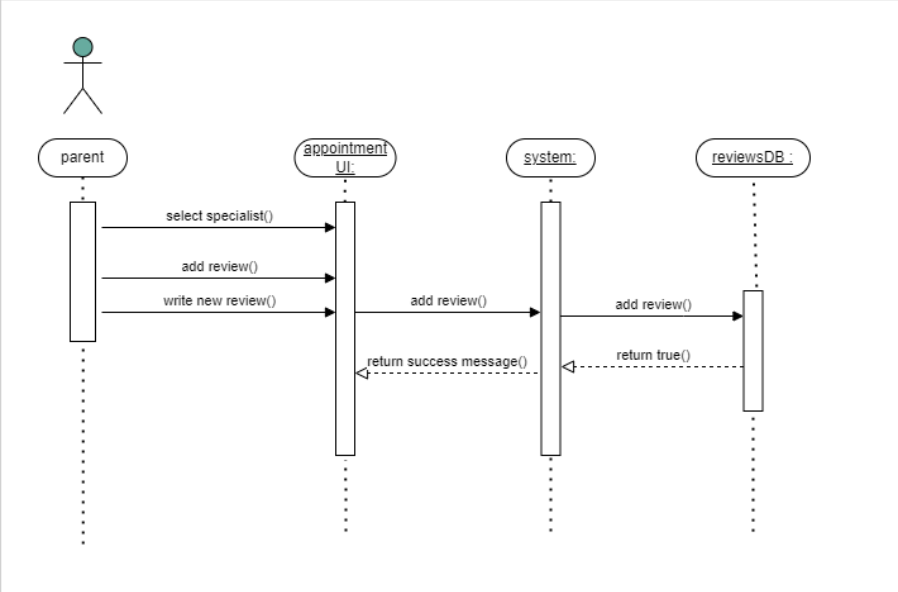
3.6.3.9 Delete child profile.



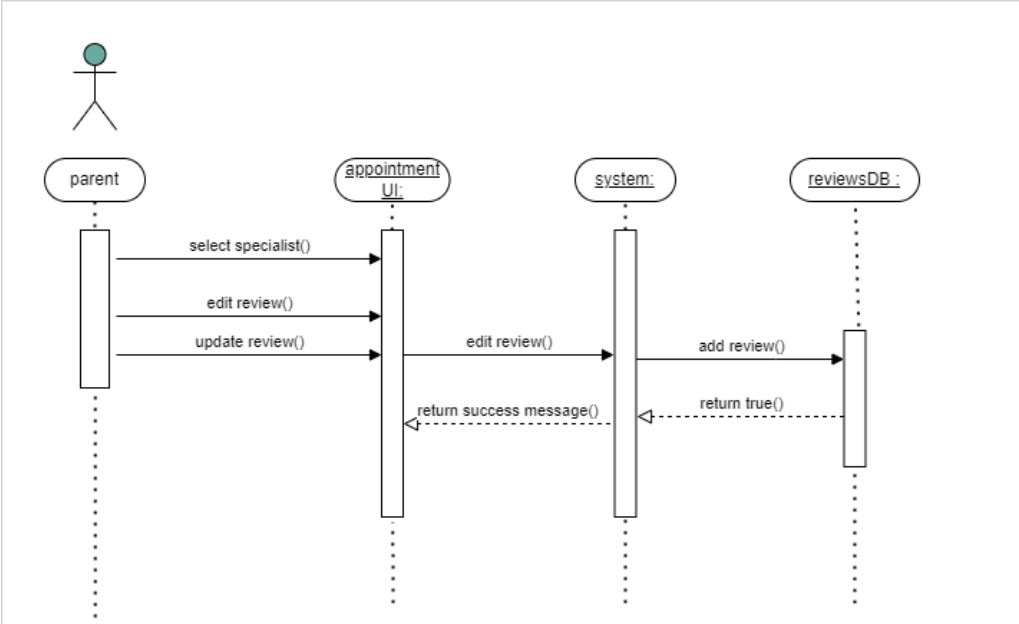
3.6.3.10 Add child.



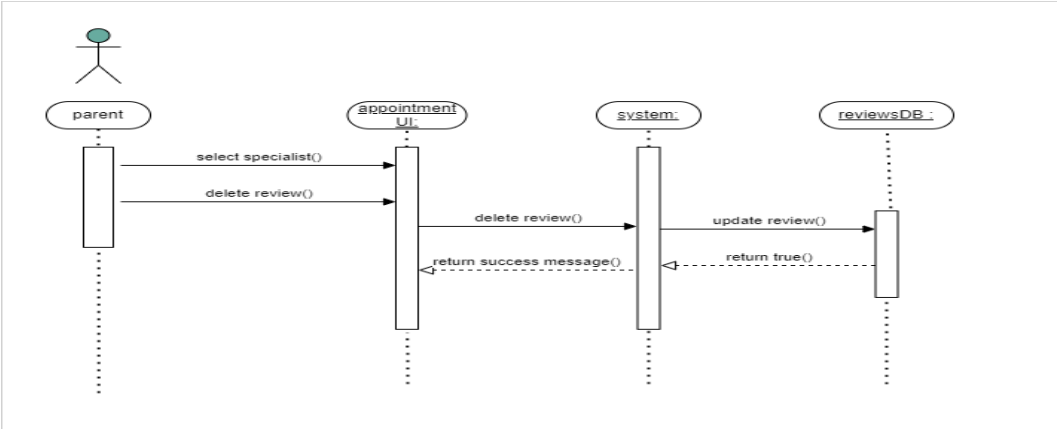
3.6.3.11 Add review.



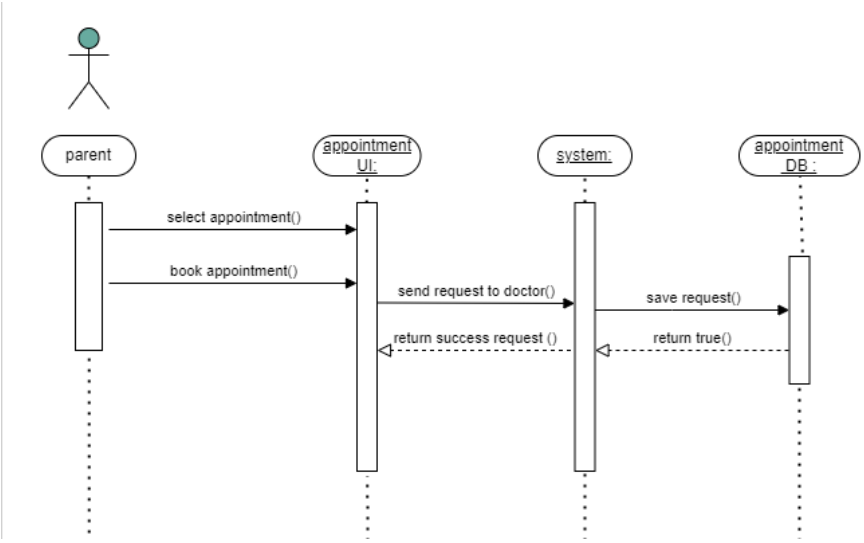
3.6.3.12 Edit review.



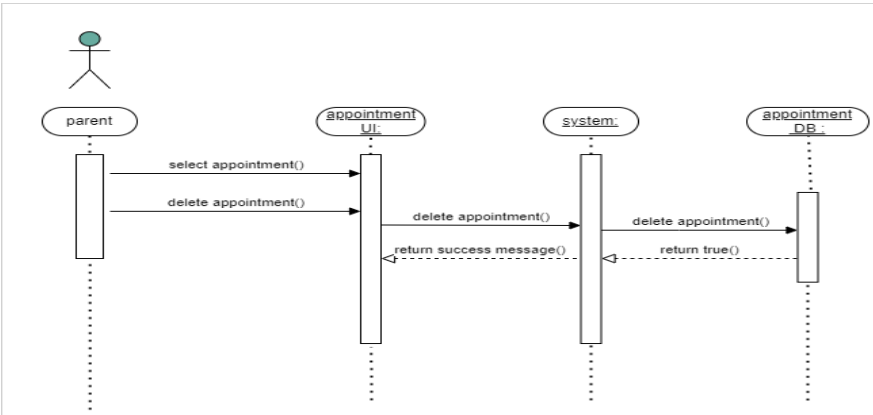
3.6.3.13 Delete review.



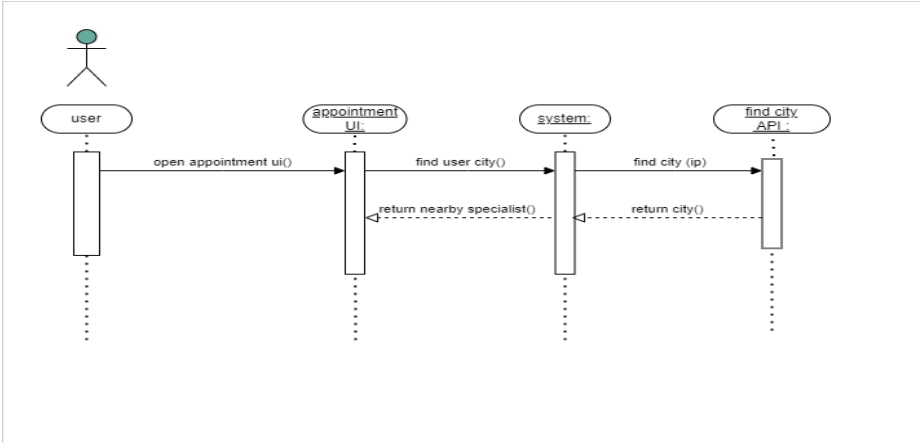
3.6.3.14 Book an appointment.



3.6.3.15 delete an appointment.



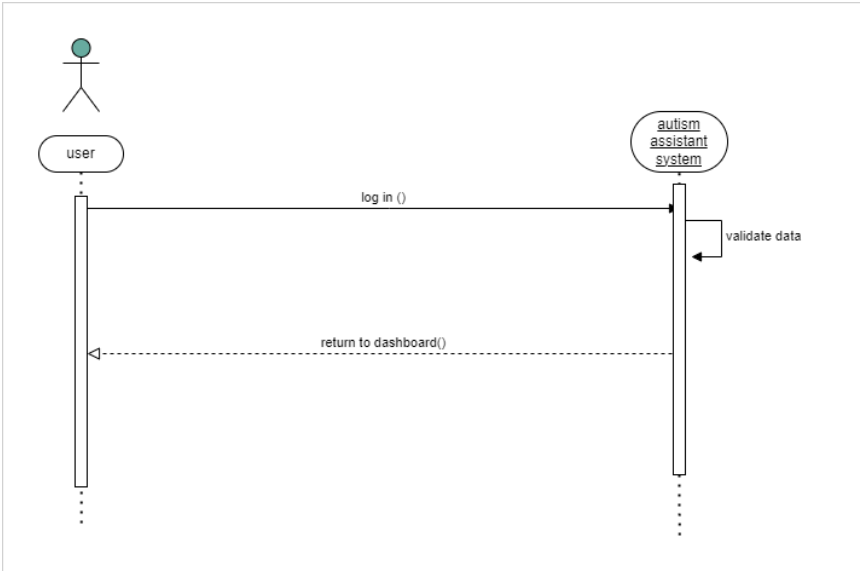
3.6.3.16 Find city.



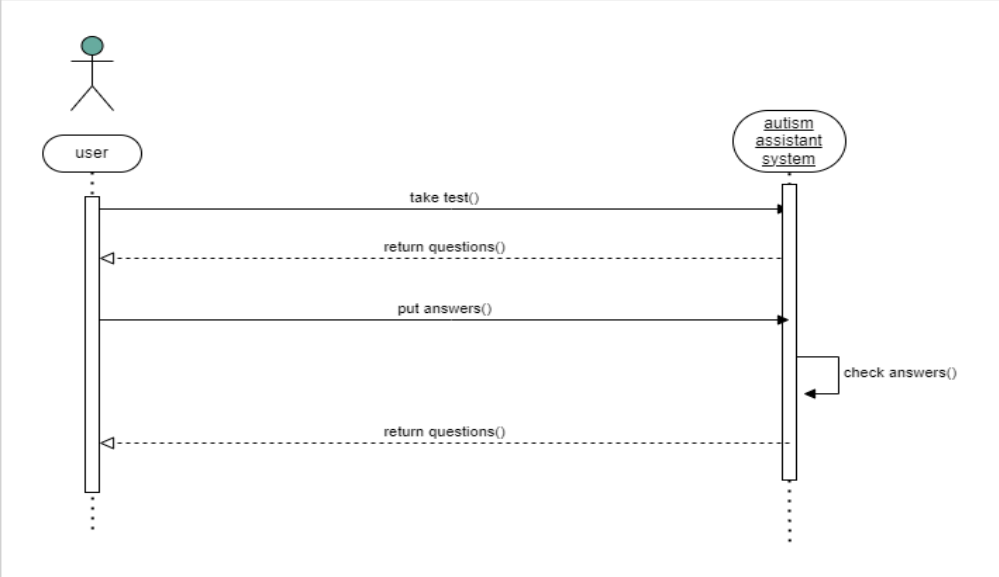
3.6.4 System sequence diagrams

System sequence diagrams (SSDs) are a type of sequence diagram that focus on the interactions between the system and external actors. SSDs show the messages exchanged between the system and external actors in a time-ordered manner. They are used to model the external behaviour of a system and can be used to identify the inputs and outputs of the system.

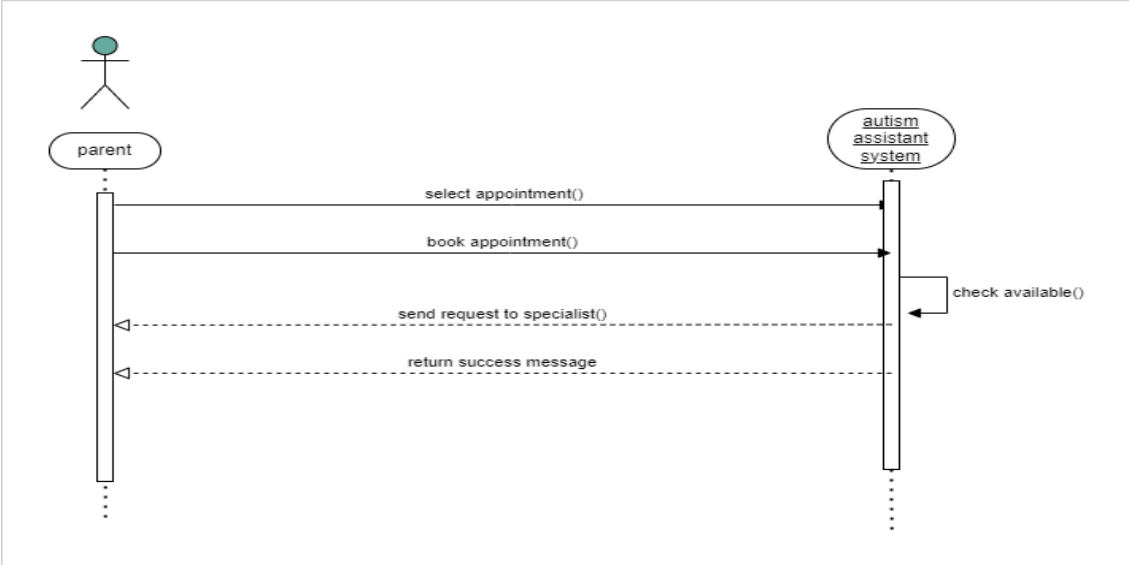
3.6.4.1 login.



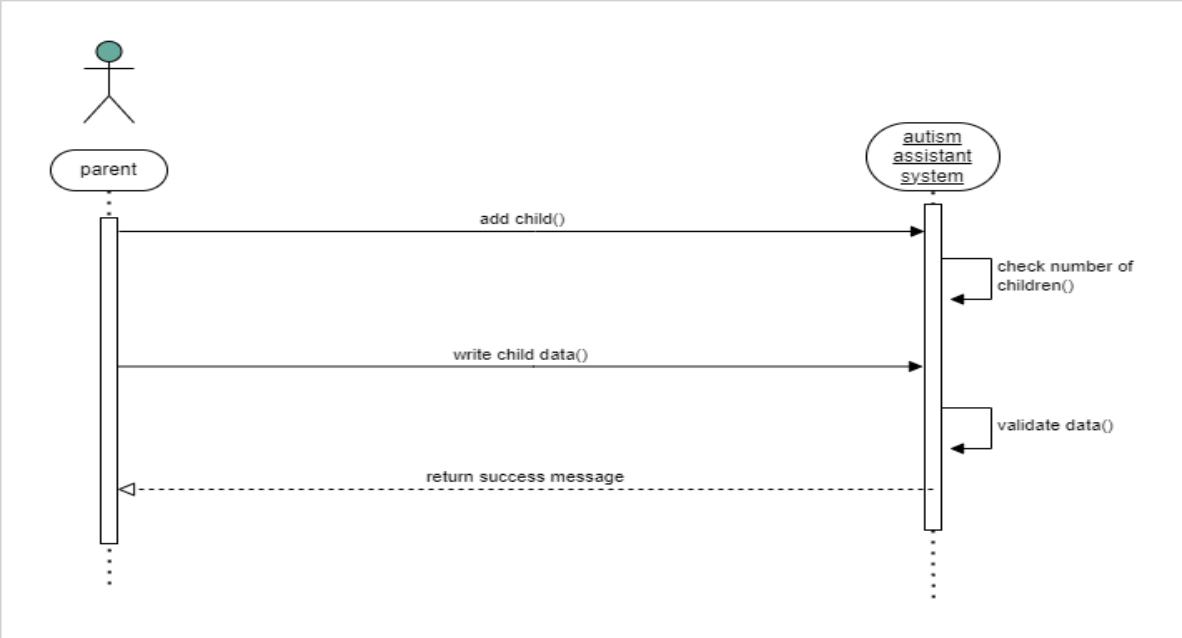
3.6.4.2 Take test.



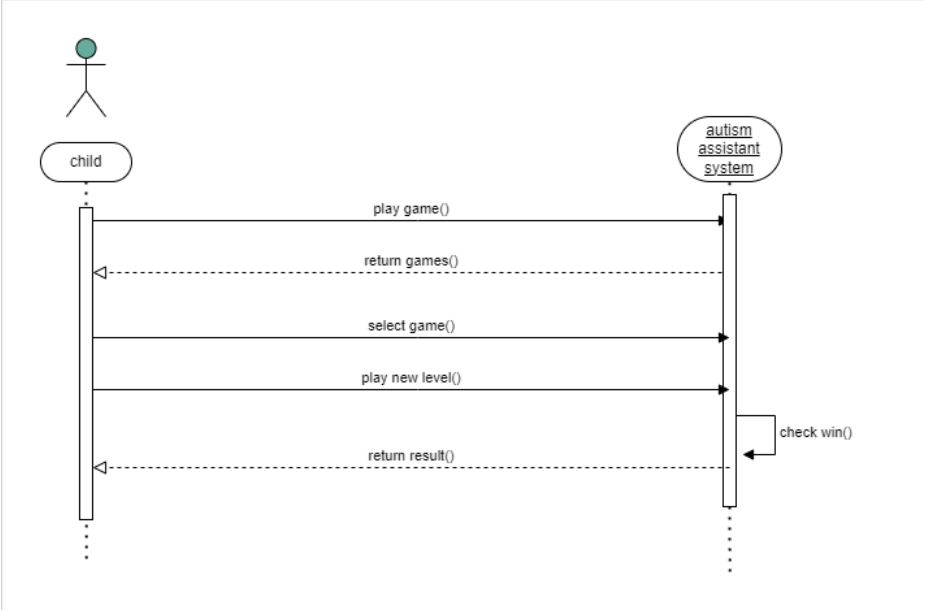
3.6.4.3 Book an appointment.



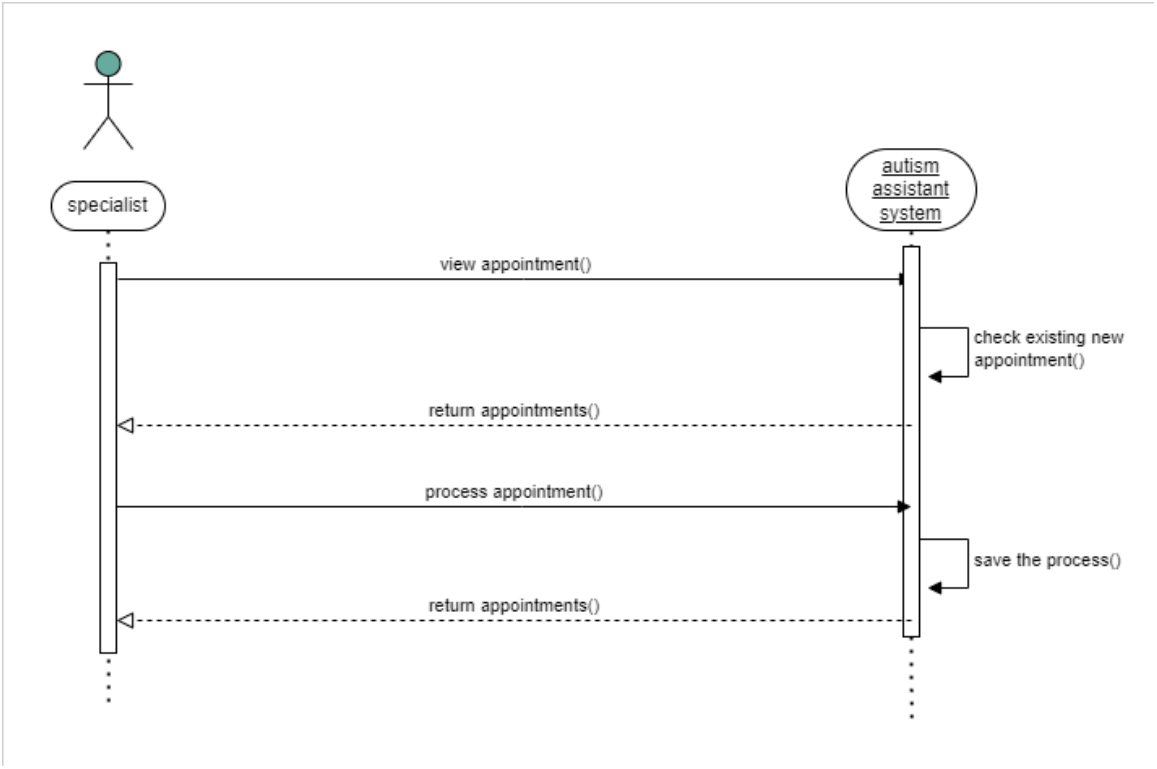
3.6.4.4 Add child.



3.6.4.5 Play game.



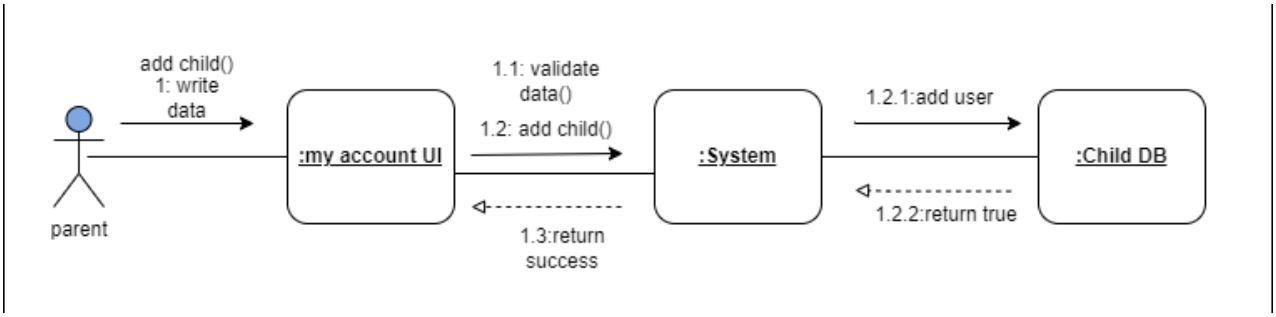
3.6.4.6 View and process appointment.



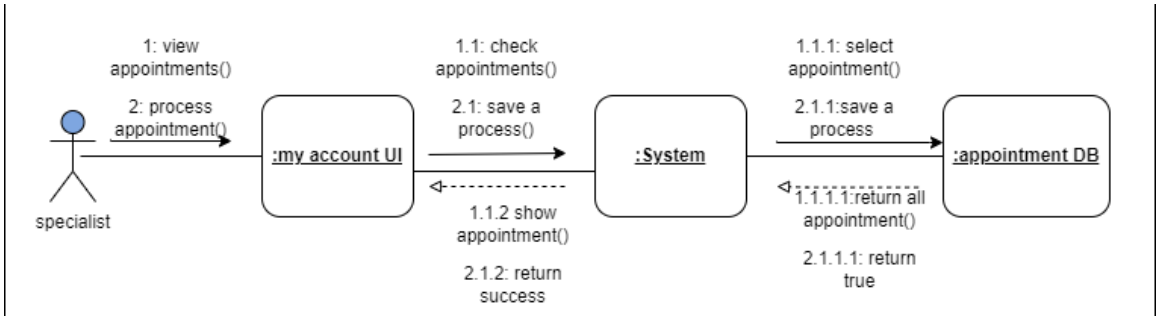
3.6.5 Collaboration diagrams

Collaboration diagrams, also known as communication diagrams, illustrate the interactions between objects or components in a system in a more abstract way than sequence diagrams. Collaboration diagrams show the relationships between objects or components and the messages exchanged between them, but they do not focus on the time ordering of messages like sequence diagrams. Collaboration diagrams are often used to model complex systems with many interacting components and to identify the dependencies between components.

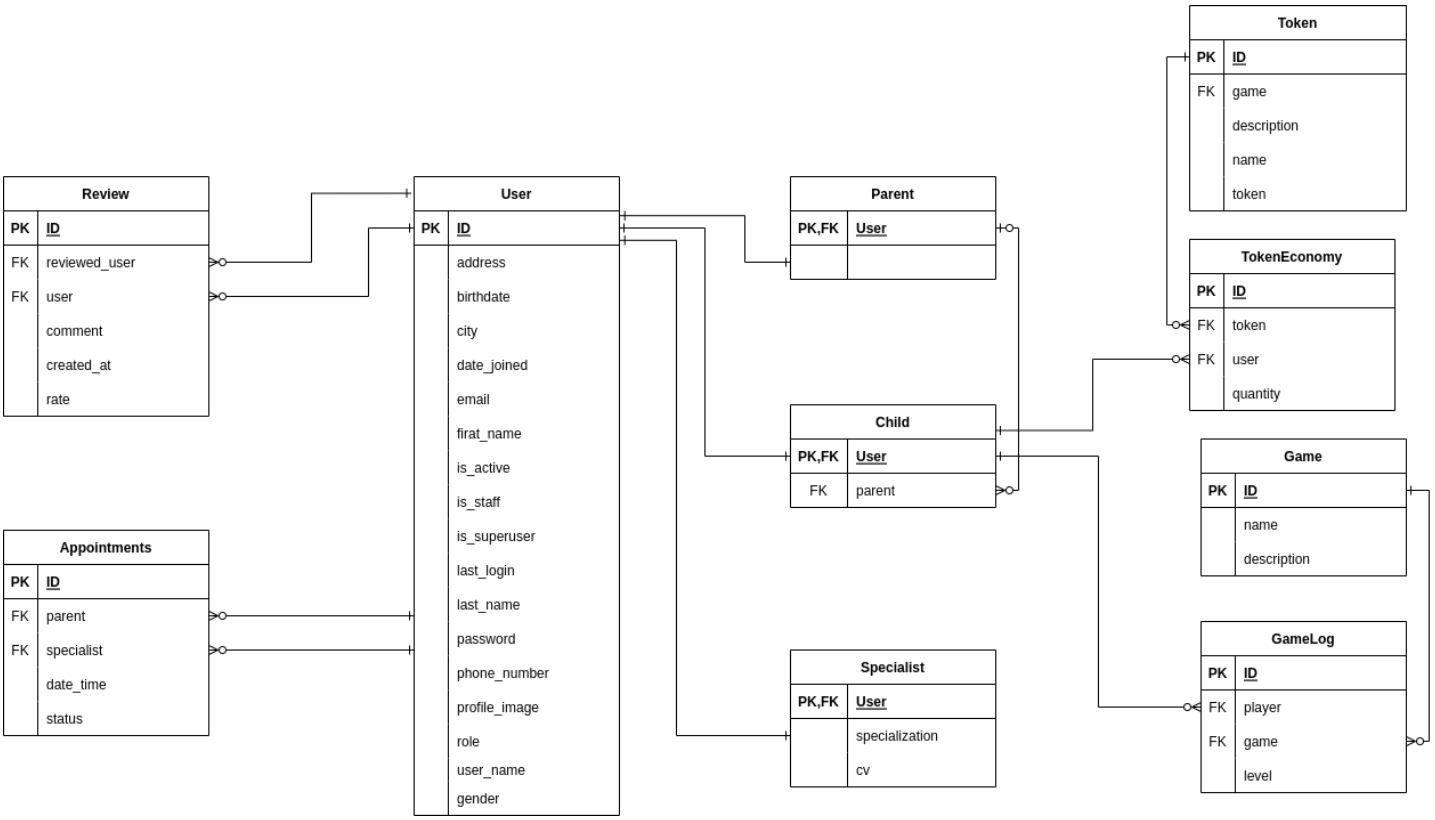
3.6.5.1 Add Child.



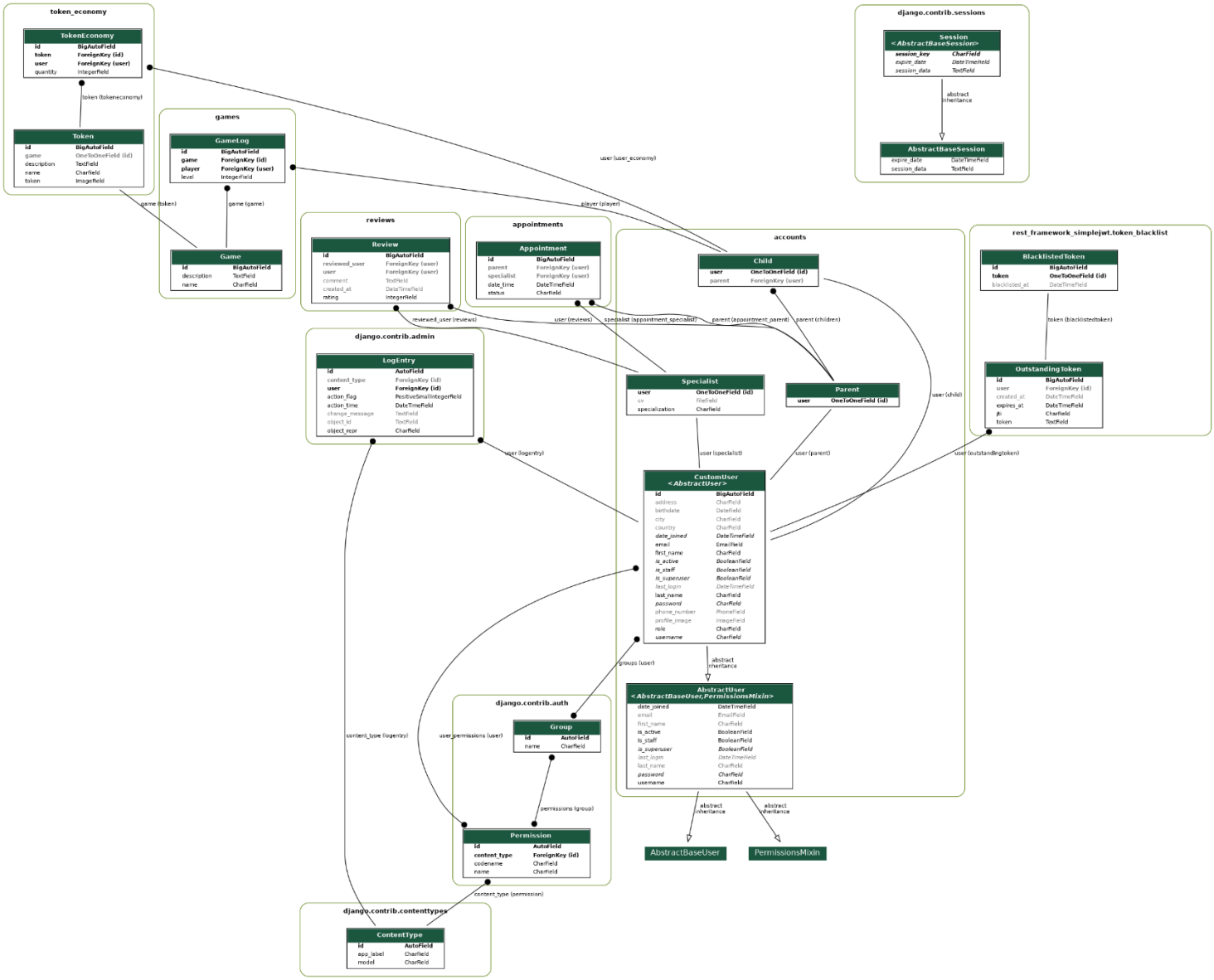
3.6.5.2 View and process appointment.



3.6.6 ERD diagram



3.6.7 Class Diagram



Chapter 4

[Implementation Details]

This chapter delves into the implementation details of our system design, highlighting the key factors that influenced our approach. With a focus on precision and clarity, we offer a comprehensive overview of the main components of our system, including the ML models, games, backend web application, and frontend web application. The chapter will also highlight the various tools and technologies used in the implementation process, and how they were integrated to achieve our objectives. Overall, our focus is on providing a comprehensive account of the implementation process, giving readers a detailed understanding of how our system was brought to life.

4.1 ML Models

4.1.1 Used tools and python packages.

4.1.1.1 Used Tools

- Kaggle
- Google Colab
- Jupyter Lab

4.1.1.2 Python packages

- NumPy
- Pandas
- Matplotlib
- Sklearn
- Keras
- Os
- Shutil
- Random
- Opencv
- MTCNN

4.1.2 Autism Screening Dataset Pre-processing

To begin, we remove null values and extraneous columns from the dataset, specifically the 'case no.', 'why taken screening', 'screening type', and 'residence' columns. This results a dataset with 19 columns, down from 24.

Next, we convert the child's text data to numerical data using label encoding. This is done for the 'Sex', 'Ethnicity', 'Jaundice', 'Family_ASD', 'Used_App_Before', 'Language', 'User', and 'Class' columns, with each unique value assigned its own number. To aid in training, we scale the numerical data to be between 0 and 1 by dividing each column by its maximum value.

Finally, we prepare the dataset for training by separating the training data from the labels and applying categorical encoding to the labels. The dataset is then divided into 60% training, 20% validation, and 20% testing, and we are ready to build our model.

4.1.3 FER Dataset Pre-processing

These following steps ensure that the input images are in the correct format and have the appropriate pixel values for optimal performance with the MobileNet architecture:

1. **Image Resizing**

MobileNet requires input images to have a specific size, typically 224x224 or 192x192 pixels. The pre-processing function resizes the input image to the required dimensions using an appropriate interpolation method, such as bilinear or bicubic interpolation. This step ensures that the input image has the correct spatial dimensions for the model.

2. **Colour Space Conversion**

MobileNet is trained on RGB images, so the pre-processing function converts the input image to the RGB colour space if it is in a different format (e.g., grayscale or BGR). This step ensures that the input image has the correct colour representation for the model.

3. **Pixel Value Scaling**

MobileNet expects input images to have pixel values in the range of [-1, 1]. The pre-processing function scales the pixel values of the input image from their original range (usually [0, 255] for 8-bit images) to the required range. This is done by dividing the pixel values by 127.5 and then subtracting 1.

This step ensures that the input image has the correct pixel value range for the model, which is important for maintaining the model's performance and generalization capabilities.

4. **Batch Dimension Addition**

MobileNet processes images in batches, so the pre-processing function adds a batch dimension to the input image. This is done by expanding the dimensions of the input image along the 0th axis.

This step ensures that the input image has the correct shape for the model, which expects a 4D tensor with dimensions (batch size, height, width, channels).

4.1.4 Autism Screening Model training and results

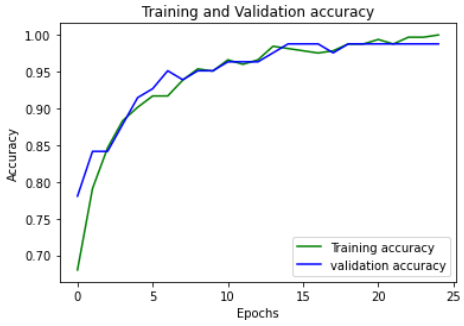
4.1.4.1 Training process

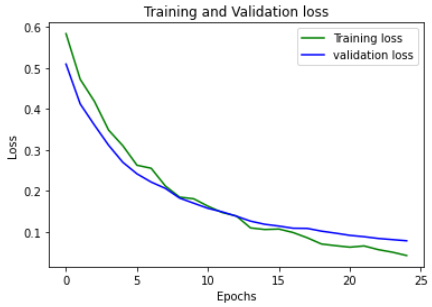
```
Epoch 13/25
41/41 [==============================] - 0s 3ms/step - loss: 0.1385 - accuracy: 0.9662 - val_loss: 0.1380 - val_accuracy: 0.9634
Epoch 14/25
41/41 [==============================] - 0s 3ms/step - loss: 0.1093 - accuracy: 0.9846 - val_loss: 0.1259 - val_accuracy: 0.9756
Epoch 15/25
41/41 [==============================] - 0s 3ms/step - loss: 0.1054 - accuracy: 0.9815 - val_loss: 0.1182 - val_accuracy: 0.9878
Epoch 16/25
41/41 [==============================] - 0s 2ms/step - loss: 0.1065 - accuracy: 0.9785 - val_loss: 0.1138 - val_accuracy: 0.9878
Epoch 17/25
41/41 [==============================] - 0s 3ms/step - loss: 0.0981 - accuracy: 0.9754 - val_loss: 0.1085 - val_accuracy: 0.9878
Epoch 18/25
41/41 [==============================] - 0s 3ms/step - loss: 0.0852 - accuracy: 0.9785 - val_loss: 0.1081 - val_accuracy: 0.9756
Epoch 19/25
41/41 [==============================] - 0s 3ms/step - loss: 0.0700 - accuracy: 0.9877 - val_loss: 0.1014 - val_accuracy: 0.9878
Epoch 20/25
41/41 [==============================] - 0s 3ms/step - loss: 0.0659 - accuracy: 0.9877 - val_loss: 0.0965 - val_accuracy: 0.9878
Epoch 21/25
41/41 [==============================] - 0s 3ms/step - loss: 0.0623 - accuracy: 0.9938 - val_loss: 0.0912 - val_accuracy: 0.9878
Epoch 22/25
41/41 [==============================] - 0s 2ms/step - loss: 0.0653 - accuracy: 0.9877 - val_loss: 0.0878 - val_accuracy: 0.9878
Epoch 23/25
41/41 [==============================] - 0s 3ms/step - loss: 0.0563 - accuracy: 0.9969 - val_loss: 0.0835 - val_accuracy: 0.9878
Epoch 24/25
41/41 [==============================] - 0s 3ms/step - loss: 0.0500 - accuracy: 0.9969 - val_loss: 0.0807 - val_accuracy: 0.9878
Epoch 25/25
41/41 [==============================] - 0s 2ms/step - loss: 0.0418 - accuracy: 1.0000 - val_loss: 0.0779 - val_accuracy: 0.9878
```

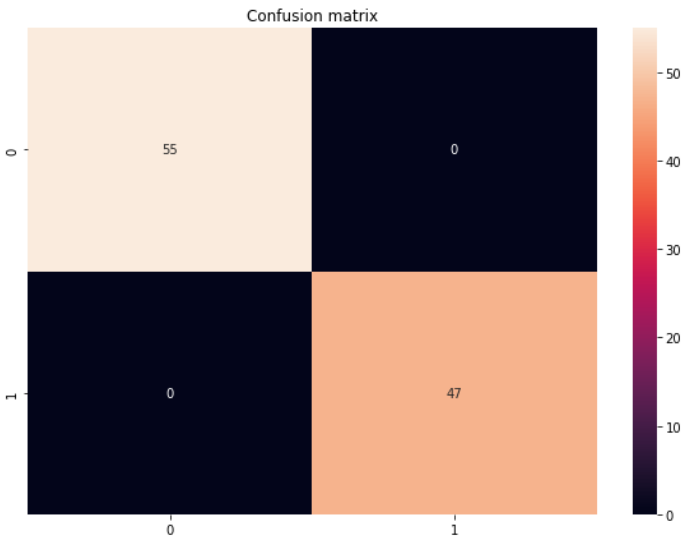
4.1.4.2 Results

The model results are 100% train accuracy, 98.78% Val accuracy and 100% testing accuracy.

Here are some plots visualize the results:







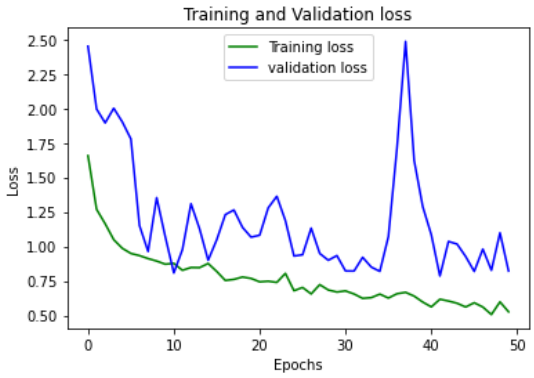
4.1.5 Fine tuning MobilNet results

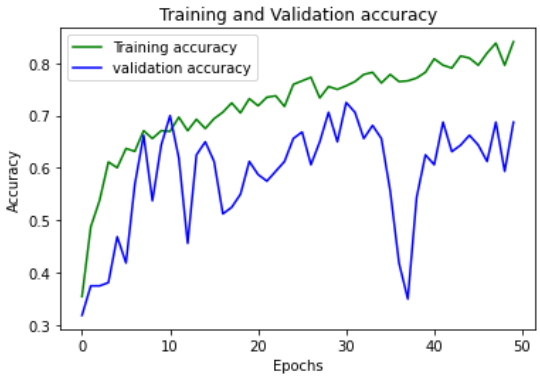
The model results are 84% train accuracy, 69% Val accuracy and 69% testing accuracy.

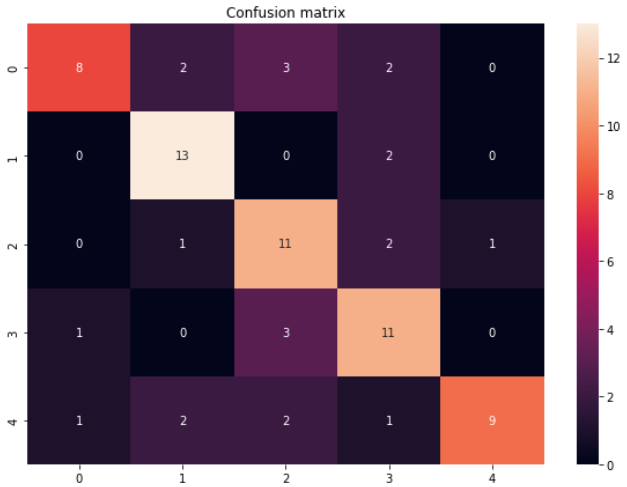
This snapshot from training process:

```
Epoch 41/50
23/23 [==============================] - 22s 970ms/step - loss: 0.5624 - accuracy: 0.8084 - val_loss: 1.0843 - val_accuracy: 0.6062
Epoch 42/50
23/23 [==============================] - 22s 975ms/step - loss: 0.6179 - accuracy: 0.7962 - val_loss: 0.7865 - val_accuracy: 0.6875
Epoch 43/50
23/23 [==============================] - 23s 987ms/step - loss: 0.6050 - accuracy: 0.7908 - val_loss: 1.0364 - val_accuracy: 0.6313
Epoch 44/50
23/23 [==============================] - 24s 1s/step - loss: 0.5890 - accuracy: 0.8139 - val_loss: 1.0174 - val_accuracy: 0.6438
Epoch 45/50
23/23 [==============================] - 23s 991ms/step - loss: 0.5618 - accuracy: 0.8098 - val_loss: 0.9274 - val_accuracy: 0.6625
Epoch 46/50
23/23 [==============================] - 24s 1s/step - loss: 0.5926 - accuracy: 0.7962 - val_loss: 0.8193 - val_accuracy: 0.6438
Epoch 47/50
23/23 [==============================] - 22s 967ms/step - loss: 0.5615 - accuracy: 0.8193 - val_loss: 0.9809 - val_accuracy: 0.6125
Epoch 48/50
23/23 [==============================] - 23s 983ms/step - loss: 0.5071 - accuracy: 0.8383 - val_loss: 0.8278 - val_accuracy: 0.6875
Epoch 49/50
23/23 [==============================] - 23s 991ms/step - loss: 0.5987 - accuracy: 0.7962 - val_loss: 1.0987 - val_accuracy: 0.5938
Epoch 50/50
23/23 [==============================] - 25s 1s/step - loss: 0.5269 - accuracy: 0.8410 - val_loss: 0.8220 - val_accuracy: 0.6875
```

And are some plots visualizing the results:







4.2 Games

4.2.1 Tools

- Freepik
- Adobe illustrator
- Picsart
- Unity Game Engine

4.2.2 Differences Game

4.2.2.1 Game structure

In this section of the documentation, we will provide an overview of the game structure, detailing the organization of scenes, prefabs, scripts, and other assets within the project.

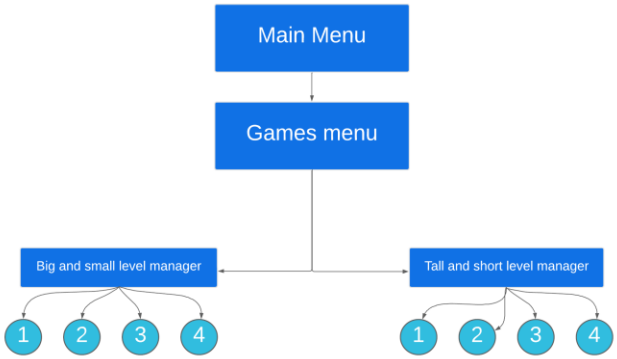
4.2.2.1.1 Folder Structure

- **Assets:** Contains all the game assets, including scripts, sprites, animations, audio, and prefabs.
- **Scenes:** Stores all the game scenes, such as the main menu, level selection, and individual game levels.
- **Prefabs:** Holds reusable game objects, such as characters, enemies, and UI elements.
- **Scripts:** Contains all the C# scripts that control game logic, character behaviour, and UI interactions.
- **Sprites:** Stores all the 2D images.
- **Audio:** Includes sound effects.
- **Animations:** Contains the animations used in the game

4.2.2.1.2 Scenes

The game is divided into several scenes, each serving a specific purpose:

- **Main Menu:** The starting point of the game, where child start playing.
- **Games menu:** Allows child to choose a concept to practice.
- **Level Manager:** Represents game levels and mange locking and unlocking individual levels.
- **Level X:** represent individual game level.



4.2.2.2 Levels manage system.

The first and second levels of the game are designed to introduce and explain individual concepts. For instance, the first level features only big objects, while the second level features only small objects. In the third level, the child is expected to understand the differences between similar objects, such as tall and short versions of the same person. The fourth level introduces entirely new objects.

The game begins with the first level, and all subsequent levels are initially locked. Once the child successfully completes a level, the next level is unlocked.

4.2.2.3 win and lose system.

The win and lose system the level starts with score 0 and 3 heart lives after each wrong answer score minus by 15 and destroy heart life until all hearts are destroyed then the level ended by lose otherwise if the level continued to end without break lose condition, then the level is passed.

4.2.2.4 Scripting

- **Button_pressed:** manage click effect of menu UI buttons.
- **ClickAndDrag:** mange dragging game objects and detect which position is there.
- **levelBtn:** manage effects of levels UI buttons which level is locked, passed, and unlocked.
- **LevelManager:** manage levels according to child's progress by locking and unlocking levels.
- **SceneNavigator:** manage navigations between game scenes.
- **ScoreManagement:** manage win & lose system and scoring.
- **SimpleJSON:** imported script file used in communication between unity webgl and our APIs requests.
- **Cage:** checks if the game object inside it in the right cage or not
- **Animal:** class manage loading game object sprites and checks its position in cage or not
- **ManageBSFinalLevel:** manage gameplay of big and small game levels 1 and 2.
- **ManageBSLevel1:** manage gameplay of big and small game levels 3 and 4.
- **ManageTallAndShort:** manage gameplay of tall and short game.
- **Person:** manage animations of person game object to move toward the correct door
- **Door:** checks if the selected door by child is correct or not and make its animations

4.2.2.5 Assets and animations

Our team has put a lot of effort into designing all the sprites in our game to be vector art. We wanted to ensure that they are not only colourful but also visually appealing to children. To achieve this, we started from scratch and designed all the necessary assets to meet our specific requirements for the game. We understand the importance of creating a visually engaging experience for our young audience, and we believe that our vector art sprites will do just that. By starting from scratch, we were able to ensure that every aspect of the sprites was tailored to our needs, resulting in a cohesive and visually stunning game.

4.2.2.6 Audios

The inclusion of audio elements serves to enhance the level of engagement between the child and the game. Specifically, the audio feature provides a spoken response to the child's answer.

4.2.3 Match & classify game.

4.2.3.1 Game structure

In this section of the documentation, we will provide an overview of the game structure, detailing the organization of scenes , scripts, and other assets within the project.

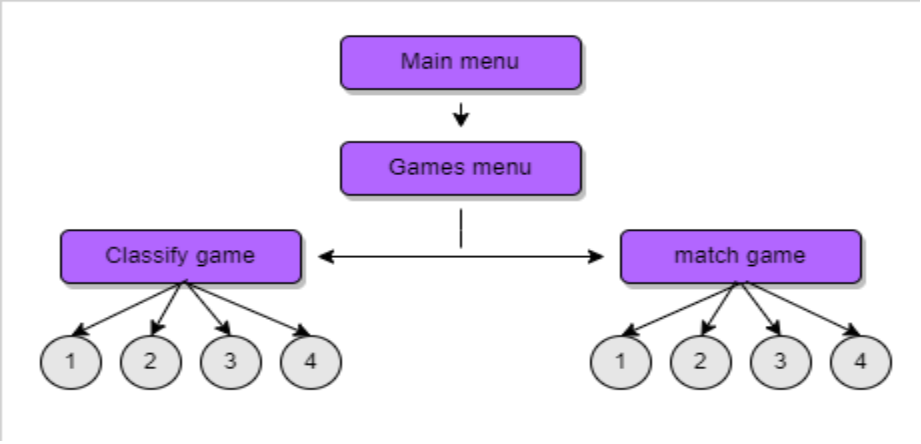
4.2.3.1.1 Folder Structure

- **Assets:** Contains all the game assets, including scripts, sprites, animations, audio.
- **Scenes:** Stores all the game scenes, such as the main menu, level selection, and individual game levels.
- **Scripts:** Contains all the C# scripts that control game logic, character behaviour, and UI interactions.
- **Simarts:** Stores all the 2D images.
- **Sounds:** Includes sound effects.
- **Animations:** Contains the animations used in the game

4.2.3.1.2 Scenes

The game is divided into several scenes, each serving a specific purpose:

- **Sample Scene:** The starting point of the game, where child start playing and Allows child to choose a concept to practice.
- **Matching levels:** The scene that has 4 matching levels .
- **Classifying levels:** The scene that has 4 matching levels .
- **Level X:** represent individual match game level.
- **Classify level X :** represent individual classify game level



4.2.3.2 Levels manage system.

The matching levels depend on match fruits, each level has number of fruits more than the previous level, the first level learns the child how to play this game ,two of four levels[level 2, level 4] have fruits that match with its shape.

The classifying levels depend on classify between more than one thing, level 1 classify between red and yellow colours, level 2 classify between cats and dogs, level 3 classify between cars and airplanes, level 4 classify between hungry time and thirsty time.

4.2.3.3 win and lose system.

The win and lose system the level starts with score 0 and 0 attempts , in two games There are an infinite number of attempts but it differs in the number of stars that the child earns from 0 to 3 stars ,but score controls the win system in classifying levels the score must be equals 4 to win the level but the match each level has its score.

4.2.3.4 Scripting

- **Level manger:** mange opening level , open new level when the child win.
- **Level transform :** transform between scenes and levels
- **Clevel manager:** manage levels of classifying game.
- **MLevel manager:** manage levels of matching game
- **Classify level X:** manage each level of classifying game and each level is differs from the other.
- **Level X:** manage each level of matching game and each level is differs from the other.
- **Get id :** manage get id of the child who plays the game and send to database to save the last level.

4.2.3.5 Assets and animations

Our team has put a lot of effort into designing all the sprites in our game to be vector art. We wanted to ensure that they are not only colourful but also visually appealing to children. To achieve this, we started from scratch and designed all the necessary assets to meet our specific requirements for the game. We understand the importance of creating a visually engaging experience for our young audience, and we believe that our vector art sprites will do just that. By starting from scratch, we were able to ensure that every aspect of the sprites was tailored to our needs, resulting in a cohesive and visually stunning game.

4.2.3.6 Audios

The inclusion of audio elements serves to enhance the level of engagement between the child and the game. Specifically, the audio feature provides a spoken response to the child's answer.

4.2.4 Emotion Classifier Game

4.2.4.1 Game Logic

The game follows a set of rules and logic to provide an interactive experience. The key aspects of the game logic are as follows:

- **Emotion Levels:** The game includes five levels of emotions: happy, sadness, anger, neutral, and surprise. Each level represents a specific emoji and adjective.
- **Timer:** A timer is set to a default value of 10 seconds at the beginning of each level. The player must mimic the emotion within this time limit.
- **Game Progression:** The player starts at level 0 (happy) and progresses to the next level upon successfully mimicking the current emotion. If the player fails to mimic the emotion within the time limit, the timer decreases by 1 second. Once the player reaches level 4 (surprise), the game ends.
- **Feedback:** The game provides visual and toast notifications to provide feedback on the player's performance. Successful mimicking prompts a success toast message, while failure to mimic within the time limit leads to a timer reduction.

4.2.4.2 Backend Integration

The game integrates with a backend server for emotion detection. The backend receives an image captured from the camera stream, processes it, and predicts the corresponding emotion using a machine learning model. The backend server is built using Django and provides an API endpoint for emotion detection.

4.2.4.3 ML Model

The machine learning model used for emotion detection is based on the FER (Facial Expression Recognition) model. The model is trained to recognize emotions such as anger, happy, neutral, sadness, and surprise.

4.2.4.4 Image Pre-processing

Before making predictions, the captured image is pre-processed to improve the model's performance. The following steps are performed:

1. **Grayscale Conversion:** The captured image is converted to grayscale to reduce colour variations.
2. **Image Enhancement:** The grayscale image is transformed into an RGB image by stacking three grayscale channels together.
3. **Face Detection:** The MTCNN (Multi-task Cascaded Convolutional Networks) algorithm is used to detect the face within the image.
4. **Cropping:** The detected face is cropped and resized to match the input size required by the emotion recognition model.
5. **Prediction:** The pre-processed face image is fed into the emotion recognition model to predict the dominant emotion.

4.2.4.5 Conclusion

The Emotion Classifier game combines real-time video processing, machine learning, and react to create an interactive experience that helps individuals recognize and mimic different emotions. By providing visual feedback and evaluation, the game aims to improve emotion recognition skills in an engaging and enjoyable manner.

4.3 Backend

4.3.1 Used tools.

- Django
- PostgreSQL
- Django rest framework
- Django rest framework-simple jwt
- Django-cors-headers
- Drf-nested-routers
- Abstract API

4.4 Frontend

4.4.1 Used tools and React JS packages.

4.4.1.1 Used Tools

- Visual Studio Code
- Git Hub

4.4.1.2 React JS packages.

- React
- React-Router-Dom
- Axios

4.4.2 Pages.

- **Home**: contains some information about Autism and some components of website
- **Appointments**: contains the list of doctors and specialists and scheduling appointments with them
- **Questionnaire**: it is a quiz that detect if child have the autism syndrome or not by sending the answers to ML models and receive the result
- **Treatments**: Contains variety of games that are specifically designed to help children with autism improve their cognitive, social, and communication skills , this games like [Differences between concepts, Matching & Classifying, Emotions]
- **About**: this page contains information about the project and what we do.
- **Contact**: this page for user if he wants to contact us or sending any message to us.
- **My Account**: this page contains personal details about the user and contains if the user wants to edit his profile.
- **Login**: this page to user to sign in into our website and if he don't have account he can create new account if he was parent or specialist or child.

Chapter 5

[conclusions]

The final chapter of this report presents a summary of our findings and draws conclusions on the effectiveness of our system. Our aim is to provide a comprehensive overview of the results obtained from our research and offer insights into how our system can be improved in the future. We will provide a detailed analysis of the data collected during the testing phase, highlighting the strengths and weaknesses of our system. In addition, we will discuss the limitations of our approach and how they can be addressed in future work.

5.1 Summary and conclusions

The Autism Care System project is a vital web application that offers a wide array of services to children with autism and their families. With its expertly designed games, the app aims to enhance cognitive, social, and communication skills in children with autism, catering to their unique needs. Furthermore, the app serves as a platform for parents to find and schedule appointments with specialized caregivers, providing detailed profiles for informed decision-making.

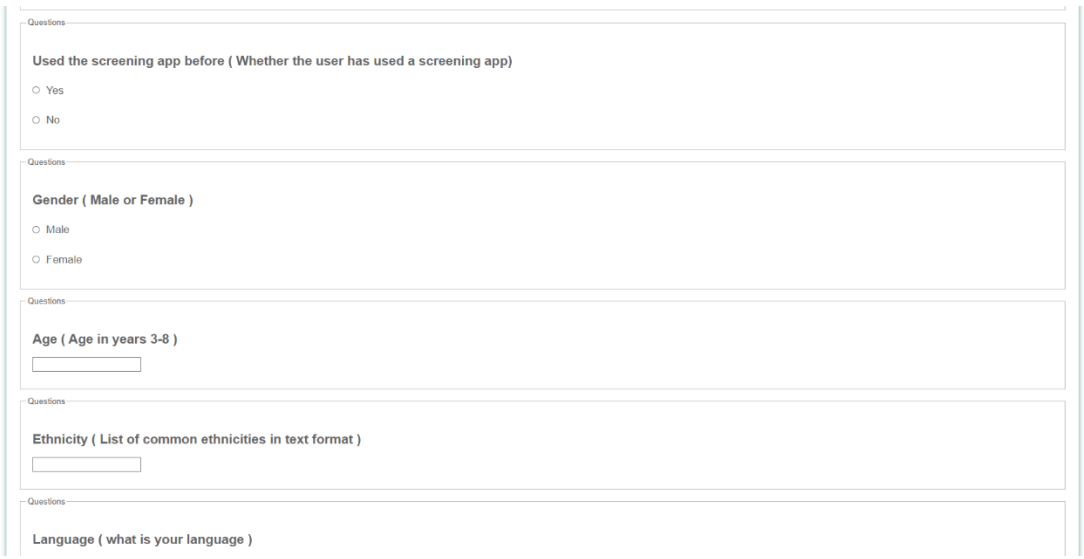
A standout feature of the Autism Care System is its machine learning-based autism detection capability, which can be an invaluable asset for parents seeking early intervention for their child's development. Ultimately, the Autism Care System is an indispensable tool for parents and caregivers dedicated to improving the lives of children with autism, offering a comprehensive range of features and services to support their needs.

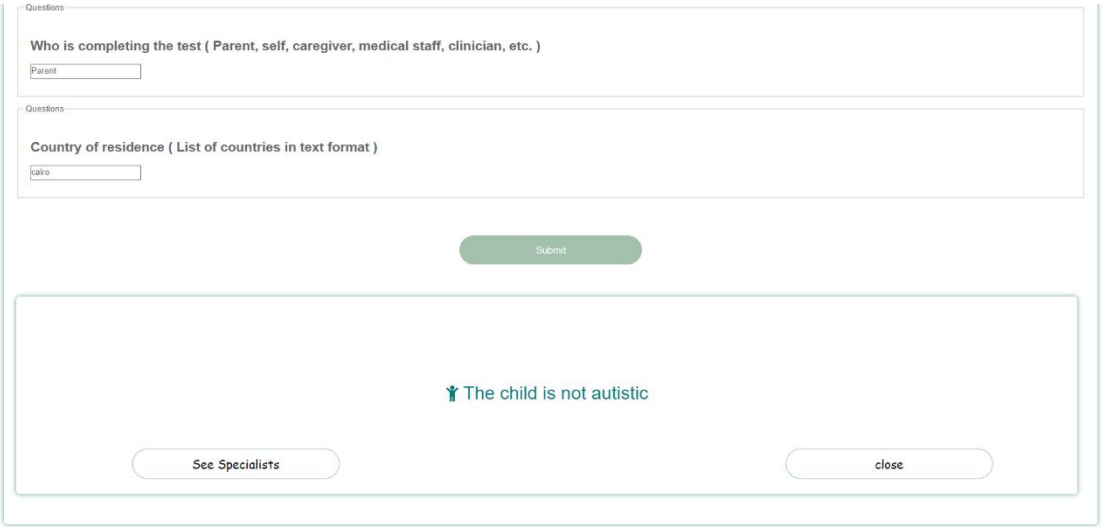
5.2 Future works

- Mobile app.
- Arabic version.
- Language skills games with NLP.
- Support for different ages.
- Academic games.
- Online meetings.
- FAQ chat bot.
- Chat subsystem.
- Develop the token economy system.
- Computer vision behaviour analysis.

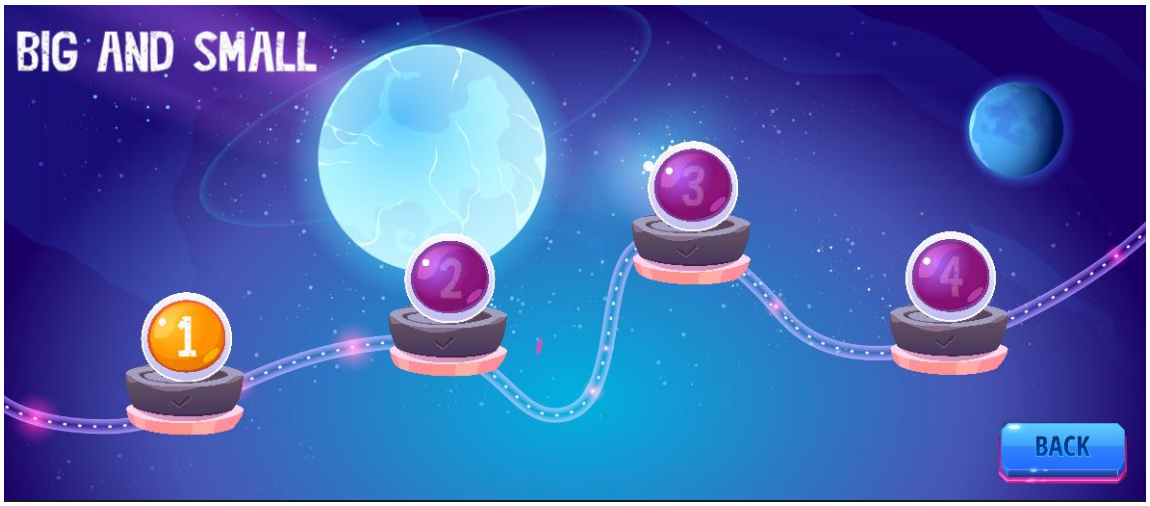
5.3 Screenshots from Applications

5.3.1 Questionnaire

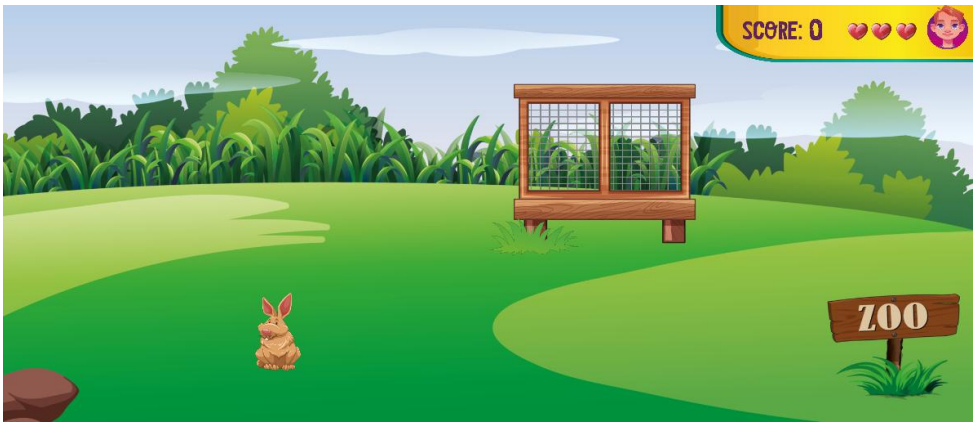




5.3.2 Differences Game

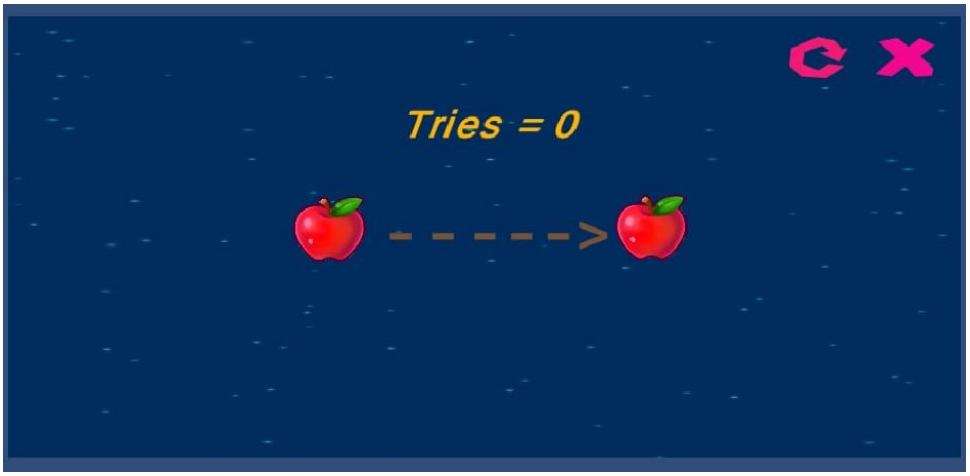


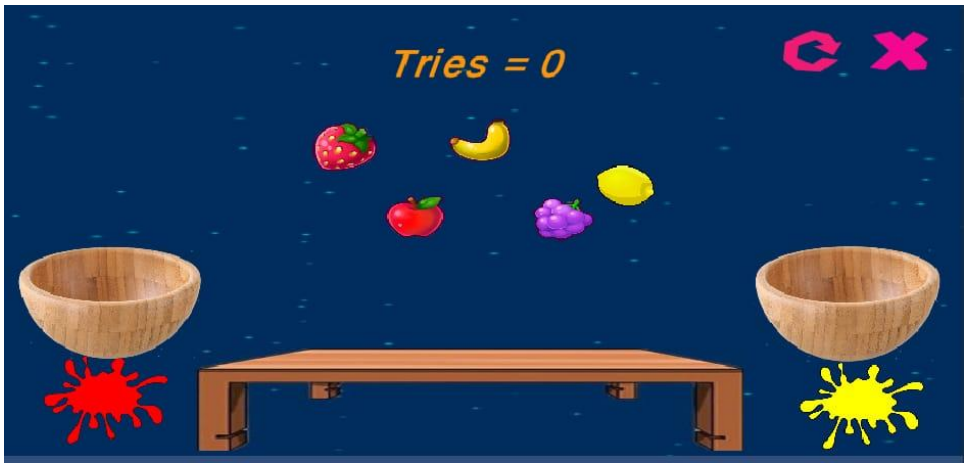

SCORE: 0
ZOO


SCORE: 0
ZOO

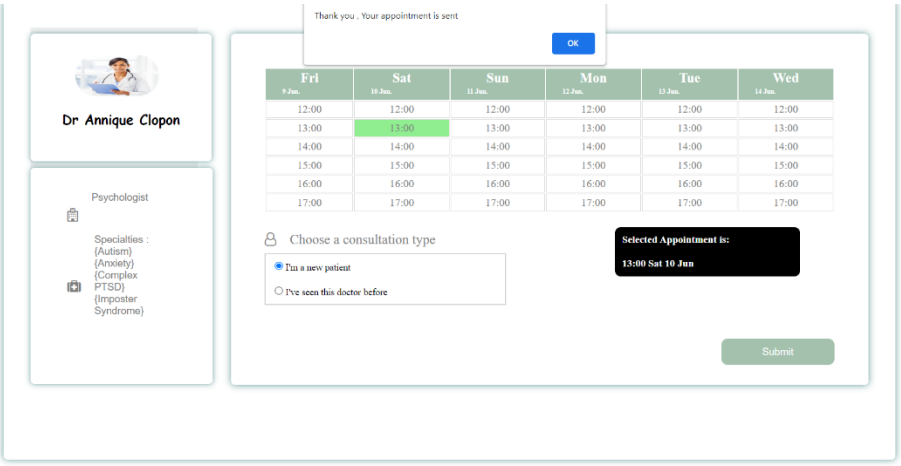

SCORE: 0

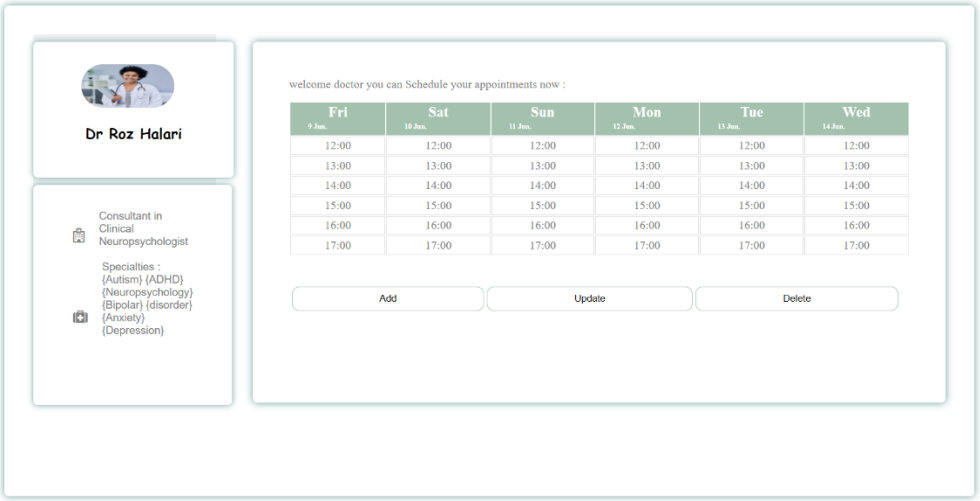
5.3.3 Classify and match Game.





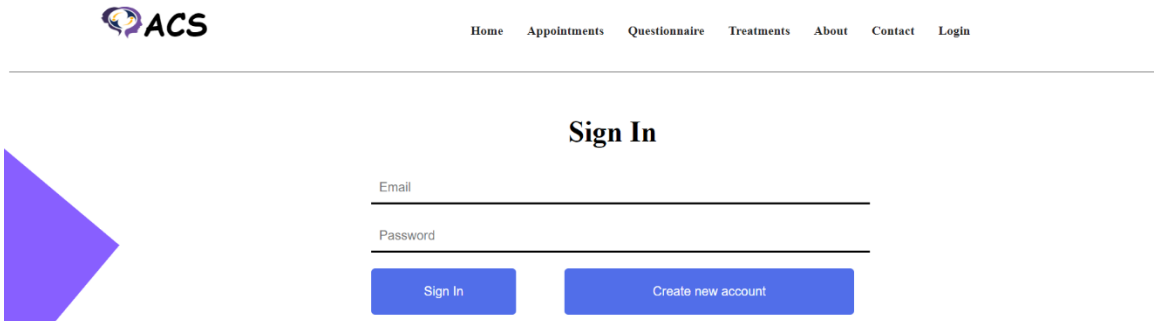
5.3.4 other screenshots from application

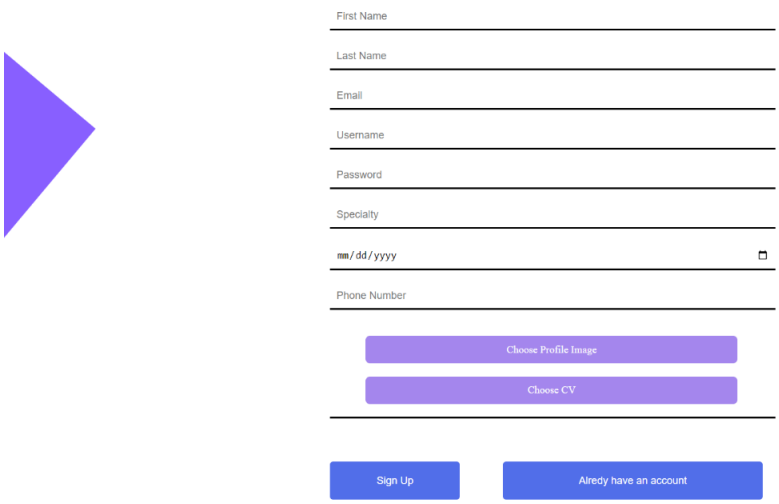


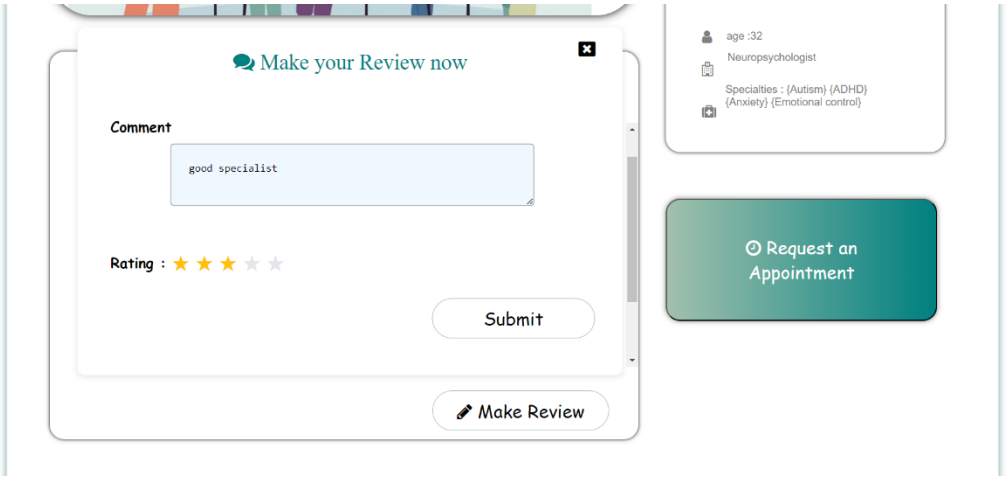


Doctor Appointment Page

Patient Name	Date	Time	Actions
Moaz Ahmed	20/5/2001	5:30 pm	Accept Decline
Karim Ahmed	28/11/2001	11:00 pm	Accept Decline
Karim Ahmed	28/11/2001	11:00 pm	Accept Decline
Karim Ahmed	28/11/2001	11:00 pm	Accept Decline
Karim Ahmed	28/11/2001	11:00 pm	Accept Decline
Neven Hamdy	25/12/2001	5:45 am	Accept Decline


ACS
Home
Appointments
Questionnaire
Treatments
About
Contact
Login
Sign In
Email
Password
Sign In
Create new account


First Name
Last Name
Email
Username
Password
Specialty
mm/dd/yyyy
Phone Number
Choose Profile Image
Choose CV
Sign Up
Alredy have an account



Parent Appointment Page

ID	Picture	Doctor Name	Date	Time	Status
1		Moaz Ahmed	20/5/2001	5:30 pm	Approved
2		Karim Ahmed	28/11/2001	11:00 pm	Pending
2		Karim Ahmed	28/11/2001	11:00 pm	Pending
2		Karim Ahmed	28/11/2001	11:00 pm	Pending
2		Karim Ahmed	28/11/2001	11:00 pm	Pending
3		Neven Hamdy	25/12/2001	5:45 am	Declined
3		Neven Hamdy	25/12/2001	5:45 am	Declined
3		Neven Hamdy	25/12/2001	5:45 am	Declined

References

[1] Towle, P.; Patrick, P. Autism Spectrum Disorder Screening Instruments for Very Young Children: A Systematic Review; Hindi Publishing Corporation: New York, NY, USA, 2016.

[2] Thabtah, F.; Abdelhamid, N.; Peebles, D. A machine learning autism classification based on logistic regression analysis. Health Inform. Sci. Syst. 2019, 7, 12

[3] Thabtah, F. Autism spectrum disorder screening: Machine learning adaptation and DSM-5 fulfillment. In Proceedings of the 1st International Conference on Medical and Health Informatics, Taichung City, Taiwan, 20–22 May 2017; pp. 1–6

[4] Wiggins, L.; Robins, D.; Bakeman, R.; Adamson, L. Brief report: Sensory abnormalities as distinguishing symptoms of autism spectrum disorders in young children. J. Autism Dev. Disord. 2009, 39, 1087–1091

[5] American Psychiatric Association. Diagnostic and Statistical Manual of Mental Disorders; American Psychiatric Publishing: Washington, DC, USA, 2000

[6] Schopler, E.; Reichler, R.; DeVellis, R. Toward objective classification of childhood autism: Childhood autism rating scale (CARS). J. Autism Dev. Disord. 1980, 10, 91–103

[7] Constantino, J. (SRS™) Social Responsiveness Scale; WPS: Torrance, CA, USA, 2005

[8] Risi, S.; lord, C.; Gotham, K.; Corsello, C.; Chrysler, C.; Szathmari, P.; Pickless, A. Combining Information from Multiple Sources in the Diagnosis of Autism Spectrum Disorders. J. Am. Acad. Child. Adolesc. Psychiatry 2006, 45, 1094–1103

[9] Rutter, M.; LeCouteur, A.; Lord, C. Autism Diagnostic Interview—Revised; WPS: Torrance, CA, USA, 2003

[10] Allison, C.; Baron-Cohen, S.; Cassidy, S.; Auyeung, B.; Lai, M.; Achoukhi, M.; Pohl, A. Attenuation of typical sex differences in 800 adults with autism vs. 3900 controls. PubMed 2014, 16, e102251

[11] Robins, D.; Fein, D.; Barton, M.; Green, J. The Modified Checklist for Autism in Toddlers: An initial study investigating the early detection of autism and pervasive developmental disorders. J. Autism Dev. Disord. 2001, 31, 131–144

[12] Lovaas, I.I. (1977), The Autistic Child, Wiley, New York, NY

[13] Charlop-Christy, M.H., Carpenter, M., Le, L., LeBlanc, L.A. and Kellet, K. (2002), "Using the picture exchange communication system (PECS) with children with autism: assessment of PECS acquisition, speech, social communicative behaviour, and problem behaviour", Journal of Applied Behaviour Analysis, Vol. 35 No. 3, pp. 213-31

[14] Baskett, C.B. (1996), The Effect of Live Interactive Video on the Communicative Behaviour in Children with Autism, University of North Carolina at Chapel Hill, Chapel Hill, NC

[15] Kanner, L. (1943), "Autistic disturbances of affective contact", Nervous Child, Vol. 2, pp. 217-50.

[16] Anibal Sólon Heinsfeld, Alexandre Rosa Franco, R. Cameron Craddock, Augusto Buchweitz, Felipe Meneguzzi, Identification of autism spectrum disorder using deep learning and the ABIDE dataset, Neuroimage: Clinical, Volume 17,2018,pp. 16-23,ISSN 2213-1582.

[17] Mujeeb Rahman, K.K.;Subashini, M.M. Identification of Autism in Children Using Static Facial Features and Deep Neural Networks. Brain Sci. 2022, 12, 94

[18] Erkan, Uğur, and Thanh, Dang N.H. "Autism Spectrum Disorder Detection with Machine Learning Methods." Current Psychiatry Research and Reviews 15, no. 4 (2020): 297-308

[19] Fitzgerald, G., Koury, K., & Mitchem, K. (2008). Research on computer-mediated instruction for students with high incidence disabilities. Journal of Educational Computing Research, 38, 201–233.

[20] Monibi, M., Hayes, G.R.: Mocotos: Mobile Communications Tools for Children with Special Needs. In: IDC 2008, pp. 121–124 (2008)

[21] Hassan, A. Z., Zahed, B. T., Zohora, F. T., Moosa, J. M., Salam, T., Rahman, M. M., Ahmed, S. I., Developing the Concept of Money by Interactive Computer Games for Autistic Children, 2011 IEEE International Symposium on Multimedia, 2011, pp. 559–564.

[22] Aggarwal, J. K., Tamersoy, B., Fernandes, T., Carlos, J., & Orvalho, V., Virtual Characters,(n.d.).

[23] Yi Li & Elmaghraby, A. S. (2014). A Framework for using Games for Behavioural Analysis of Autistic Children. The 19th International Conference on Computer Games, Louisville, Kentucky, USA, 2014

[24] Matson, J., & Boisjoli, J. (2009). The token economy for children with intellectual disabilities and/or autism: A review. Research in Developmental Disabilities., 30(2), 240-248.

[25] Alexis Bosseler, Dominic W. Massaro. (2003).Development and evaluation of a computer-Animated Tutor for Vocabulary and Language Learning in Children with Autism

[26] Canedo, Daniel, and António J. R. Neves. 2019. "Facial Expression Recognition Using Computer Vision: A Systematic Review" *Applied Sciences*.

[27] Viola, P.; Jones, M.J. Robust real-time face detection. Int. J. Comput. Vis. 2004, 57, 137–154.:VISI.0000013087.49260.fb.

[28] Kazemi, V.; Sullivan, J. One millisecond face alignment with an ensemble of regression trees. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; 2014; pp. 1867–1874.

[29] Zhang, K.; Zhang, Z.; Li, Z.; Qiao, Y. Joint face detection and alignment using multitask cascaded convolutional networks. IEEE Signal Process. Lett. 2016, 23, 1499–1503.

[30] Farfade, S.S.; Saberian, M.J.; Li, L.J. Multi-view face detection using deep convolutional neural networks. In Proceedings of the 5th ACM on International Conference on Multimedia Retrieval, Shanghai, China, 23–26 June 2015; ACM: New York, NY, USA, 2015; pp. 643–650