# Development of Android-based Applications for a Social Robot to Enhance Multicultural Inclusivity in Early Childhood Education

Abdul Rauf

DIBRIS - Department of Computer Science, Bioengineering, Robotics and System Engineering

University of Genova

Supervisors:
Prof. Carmine Recchiuto

In partial fulfillment of the requirements for the degree of

*Laurea Magistrale in Robotics Engineering*

October 15, 2024

# Declaration of Originality

I, Abdul Rauf, hereby declare that this thesis is my own work and all sources of information and ideas have been acknowledged appropriately. This work has not been submitted for any other degree or academic qualification. I understand that any act of plagiarism, reproduction, or use of the whole or any part of this thesis without proper acknowledgment may result in severe academic penalties.

# Acknowledgements

I would like to express my sincere gratitude to my supervisor, Prof. Carmine Recchiuto, for his invaluable guidance, support, and encouragement throughout the duration of this project. His expertise and insights were crucial to the development and completion of my thesis. I am truly grateful for his patience and for pushing me to achieve my best.

This is a short, optional, dedication. To all the Master and PhD students of Robotics Engineering at the University of Genova.

# Abstract

Integrating social robotics in educational settings revolutionises how children learn, engage, and interact within classrooms, especially in culturally diverse environments. Robots in schools are no longer just technological tools; they are interactive companions that support the development of language skills, cognitive abilities, and emotional intelligence. By interacting with children through multilingual communication, cultural sensitivity, and dynamic responses, robots create inclusive and engaging learning experiences that cater to diverse learning needs. Their role extends beyond traditional teaching methods by fostering active participation, social inclusion, and personalised learning, which proves more effective than passive digital devices like tablets. In intercultural pedagogy, social robots serve as cultural mediators, helping students from various backgrounds connect, share, and learn from one another. Their ability to engage with children in multiple languages and introduce them to different cultural traditions, games, and stories enhances cultural awareness and social integration. This strengthens linguistic and cognitive development and promotes empathy and mutual respect among students. Robots actively participate in symbolic games and discussions on cultural heritage and offer personalised feedback, fostering curiosity, creativity, and collaboration within diverse classrooms. The significance of social robotics in education lies in its potential to transform learning environments into inclusive, interactive, and culturally responsive spaces. Robots play a critical role in supporting holistic development by addressing both academic and social dimensions of learning. Their ability to facilitate linguistic and intercultural learning underscores the importance of incorporating robotics into modern educational practices, offering children a more equitable, inclusive, and enriching educational experience, regardless of their cultural or linguistic backgrounds.

# Contents

# List of Figures

# Chapter 1

# Introduction

## 1.1 General Context

In today's increasingly interconnected world, educational institutions must adapt to diverse cultural, linguistic, and social landscapes. Schools now host students from various backgrounds, bringing unique perspectives and learning needs (1). Creating cohesive educational experiences that cater to these diverse populations has become a critical challenge in such environments. Intercultural pedagogy, a teaching approach that fosters understanding, respect, and collaboration among students from different cultural backgrounds, has emerged as a vital strategy for addressing these challenges (2). Therefore, by focusing on inclusivity and equitable learning opportunities, intercultural pedagogy ensures that students from all backgrounds can succeed, regardless of language barriers or cultural differences.

In recent years, technology has been increasingly incorporated into education to address the complexities of multicultural classrooms. One such technological innovation is social robotics, which offers promising potential for enhancing intercultural learning experiences. These robots can interact meaningfully with children in educational settings, facilitating an immersive learning environment that supports the practice of language skills, cultural exchange, and group activities (3). The interactive nature of these robots allows for a more dynamic learning experience, particularly in the context of intercultural education. For example, robots can be programmed to engage with students in multiple languages, bridging the gap for students who may not fluently speak the language of instruction (4). Moreover, social robots can simulate social interactions and act as cultural mediators, introducing students to different cultural traditions, customs, and perspectives in an engaging and non-threatening way.

This study explores the role of social robots in supporting intercultural pedagogy, focusing on how robots can foster cultural awareness, linguistic development, and social inclusion among students (5). Through the examination of three specific scenarios in which robots interact with children in educational environments, the study highlights the transformative potential of social robotics in creating more inclusive and engaging learning experiences. It aims to contribute to the increasing social robotics into educational practices, particularly within the framework of intercultural pedagogy (6). One of the key advantages of using social robots in intercultural pedagogy is their ability to foster cultural awareness playfully and interactively (7). In diverse classrooms, where students may come from various cultural backgrounds, social robots can introduce concepts of cultural diversity through interactive storytelling, games, and role-playing. For instance, a robot could engage students in activities exploring cultural customs or celebrating worldwide festivals (8). By participating in these activities, students learn about different cultures and develop a deeper appreciation for the richness of cultural diversity. Such engagement is crucial for building intercultural competence, which is understanding and navigating cultural differences with sensitivity and respect.

In addition to promoting cultural awareness, social robots also play a vital role in supporting linguistic development. Language is often a significant barrier for students from different cultural backgrounds, particularly in schools where the language of instruction differs from their native language (9). Social robots can be programmed to interact with students in multiple languages, allowing them to practice their language skills in a supportive and non-judgmental environment (10). This can be particularly beneficial for students learning a second language or needing additional support in developing their language proficiency. Moreover, by engaging in conversations with the robots, students can build confidence in using a new language, enhancing their overall academic performance and social integration. One of the challenges of intercultural pedagogy is ensuring that students from diverse backgrounds feel included and valued in the learning process (11). Social robots can interact with students personally and help create a more inclusive classroom environment. For example, a robot can facilitate group activities that encourage collaboration among students from different cultural backgrounds, fostering teamwork and mutual understanding (12). By acting as a neutral and non-threatening facilitator, the robot can help break down social barriers and create a more cohesive classroom dynamic (13).

Moreover, social robots can be designed to address specific challenges that arise in diverse educational settings. For example, in scenarios where cultural misunderstandings or conflicts may occur, a robot could act as a mediator, helping students navigate these situations by encouraging open dialogue and mutual respect (14). This capability is significant in multicultural classrooms, where students may need support resolving cultural differences and building positive peer relationships. This way, social robots create a learning environment where all students feel supported, respected, and included. This shows that integrating social robotics into intercultural pedagogy represents a significant step forward in addressing the challenges of educating diverse student populations. By fostering cultural awareness, supporting linguistic development, and promoting social inclusion, social robots can transform how intercultural education is delivered. As schools become diverse, innovative technologies such as social robotics will play an increasingly important role in ensuring that all students have access to equitable and inclusive learning experiences. This study will further explore these possibilities by examining specific scenarios in which social robots interact with children in culturally diverse classrooms, shedding light on the potential of social robotics to enhance intercultural pedagogy in the 21st century.

## 1.2 Importance of Intercultural Pedagogy

Intercultural pedagogy is an educational approach designed to recognise and harness the richness of cultural diversity within the classroom (15). Intercultural pedagogy aims to develop an environment where students are exposed to various cultural perspectives and encouraged to reflect on their own cultural identities critically (16). This reflective process is crucial for developing mutual respect, empathy, and student collaboration, which is essential for creating cohesive and inclusive learning environments. In contemporary classrooms, which are increasingly multicultural, the need for intercultural pedagogy has never been more evident (17). With global migration, demographic changes, and the movement of people across borders, classrooms are filled with students from various cultural, ethnic, and linguistic backgrounds (18). While this diversity offers significant opportunities for enriching the educational experience, it also presents challenges that educators must navigate carefully. Students from immigrant, refugee, or minority backgrounds often encounter barriers to full participation in the classroom, ranging from linguistic difficulties to cultural misunderstandings (19). These challenges can contribute to feelings of alienation, social exclusion, or discrimination, negatively impacting academic engagement and performance.

Intercultural pedagogy responds to these challenges by equipping students with the skills and understanding necessary to navigate cultural differences (20). This approach goes beyond simply teaching about other cultures; it aims to create a classroom culture that values diversity as an asset. By promoting critical reflection on personal and collective cultural identities, intercultural pedagogy encourages students to appreciate the uniqueness and commonality of human experiences across different cultural contexts (21). This pedagogy seeks to break down stereotypes, challenge preconceived notions, and nurture a deep sense of empathy and respect among students, preparing them to interact successfully in increasingly multicultural societies (22). A key aspect of intercultural pedagogy is its emphasis on inclusivity. Rather than marginalising students who may differ from the dominant cultural norms, this approach strives to include and celebrate their contributions (23). It is particularly relevant in addressing the needs of students who may feel excluded from the mainstream educational experience due to linguistic or cultural barriers. Intercultural pedagogy ensures that all students feel valued and recognised regardless of their background (24). Educators can create an environment where students from all backgrounds can thrive by incorporating diverse cultural perspectives into the curriculum and encouraging cross-cultural dialogue.

One of the most important objectives of intercultural pedagogy is the development of cross-cultural communication skills (25). These skills are increasingly necessary in a world where individuals must navigate interactions with people from various cultural backgrounds, both personally and professionally (26). In the classroom, cross-cultural communication skills help students engage in meaningful dialogue with their peers, work collaboratively on group projects, and participate in discussions that include multiple cultural perspectives (22). These interactions enrich the educational experience and prepare students for life beyond the classroom, equipping them with the skills they will need to succeed in a globalised world. However, as societies become more diverse, educators are called to address the specific challenges students from different cultural backgrounds face.

In many cases, students from minority or immigrant backgrounds struggle to adapt to the cultural norms of their new educational environment. For instance, linguistic barriers may prevent students from fully participating in classroom discussions or understanding the curriculum (20). Cultural differences in communication styles, learning preferences, or social behaviour may also create misunderstandings between students, teachers, or peers (23). These challenges can lead to feelings of isolation and alienation, further exacerbating educational inequalities. Intercultural pedagogy provides a framework for addressing these issues by promoting inclusive teaching practices. It encourages educators to adopt teaching strategies acknowledging and accommodating cultural differences (15). For example, teachers may incorporate bilingual education programs or culturally responsive teaching materials that reflect the diverse backgrounds of their students (26). By doing so, they create a more inclusive curriculum that not only supports the academic success of minority and immigrant students but also raises a deeper understanding of cultural diversity among all students. In addition to supporting students from diverse cultural backgrounds, intercultural pedagogy is critical in promoting social cohesion within the classroom (25). In many multicultural classrooms, students from different cultural groups may form separate social circles, leading to segregation within the classroom. This segregation can prevent students from engaging with peers from different backgrounds and hinder the development of cross-cultural understanding (20). Intercultural pedagogy addresses this issue by encouraging activities that promote collaboration and interaction between students from different cultural groups. For example, group projects or class discussions that require students to work together on tasks that explore cultural differences can help break down social barriers.

Technology, specifically social robotics, has emerged as a promising tool in supporting intercultural pedagogy (17). Social robots are designed to engage with humans in social contexts and are equipped with features such as speech recognition, facial expressions, and culturally adaptable behaviours (16). These robots can play a significant role in enhancing intercultural learning by interacting with students in multiple languages, presenting culturally relevant content, and encouraging collaboration among diverse groups of students. As a result, social robots have the potential to support the development of cross-cultural communication skills and promote social inclusion in the classroom (18). One of the key benefits of using social robots in intercultural pedagogy is their ability to provide personalised and adaptive learning experiences (19). Unlike traditional teaching tools, social robots can interact with students one-on-one, adapting their responses based on each student's cultural background, language proficiency, and learning needs (21). For instance, a robot could switch between languages depending on the student's needs, offering explanations or instructions in the student's native language to facilitate understanding. This personalised approach can be particularly beneficial for students who may struggle with linguistic barriers or need additional language development support. Furthermore, social robots can serve as cultural mediators, introducing students to new cultural perspectives and encouraging them to reflect on their cultural identities (24).By facilitating group activities that require teamwork and cross-cultural communication, robots can help create an inclusive classroom environment where all students feel valued and respected (26). These interactions not only support academic learning but also contribute to students' social and emotional development, helping them build relationships across cultural lines.

## 1.3 Objectives of the Study

1. To facilitate linguistic development and cultural awareness through multilingual engagement with songs, visuals, and greetings.
2. To promote cultural exploration and dialogue through symbolic games and discussions of food traditions.
3. To encourage cultural exchange and reflection using imagined worlds and symbolic objects for a playful learning experience.

## 1.4 Structure of the Thesis

The thesis begins with an Introduction outlining the general context of multicultural inclusivity in education, discussing migratory phenomena, and highlighting the importance of intercultural pedagogy. The State of the Art reviews current use of robots in schools, with a focus on their application in intercultural pedagogy. The Methodology details the development process of Android-based applications, their integration with the Buddy robot using the Buddy SDK, and the design of key components, including robot movement, multilingual speech, and the use of the robot's tablet for images and videos. The Experiment section describes the implementation of three scenarios (Multilingual Engagement, Cultural Exploration, and Cultural Exchange) using the developed apps. Finally, the Conclusions summarize the study's findings, discussing the implications of social robots in enhancing multicultural inclusivity and proposing directions for future research.

# Chapter 2

# State of the Art

This chapter reviews the current state of social robotics in education, particularly in primary and early childhood settings. It examines how social robots have transformed traditional teaching into more student-centered, interactive experiences. The chapter discusses their role in individualized learning, collaboration, and language acquisition, contrasting these benefits with potential limitations, such as reduced human interaction and emotional sensitivity. Additionally, it explores the application of social robots in fostering intercultural awareness and inclusivity. The chapter concludes by highlighting relevant projects and ethical considerations in using robots within multicultural classrooms, providing a critical foundation for understanding their impact on modern education.

## 2.1 Robots in Education

### 2.1.1 The Role of Social Robots in Education

Introducing social robots in educational settings has revolutionised traditional learning modes, shifting from teacher-centred approaches to more student-centred, interactive learning environments (27). Traditionally, the role of the teacher was dominant, and students were often passive recipients of information. However, with the advent of social robots, learning has become more dynamic, with robots facilitating educational activities that span language acquisition, cognitive development, and social skills (28). This shift has been particularly beneficial in early childhood and primary education, where the need for engagement and tailored learning approaches is high. Social robots can provide students a more engaging and stimulating environment through advanced capabilities, such as facial recognition, voice interaction, and question-answering abilities (29). (30) has shown that the presence of robots in classrooms can profoundly impact student engagement and motivation. For instance, studies suggest students are more likely to participate and remain focused when learning through robots due to their novelty and interactive features.

One of the primary advantages of social robots in education is their ability to facilitate individualised learning (31). This is particularly beneficial when students have different learning styles, paces, or special needs. For example, some students may benefit from visual or auditory stimuli, while others might require a more hands-on or kinesthetic approach (32). (33) showed that robots are especially effective in supporting students with learning difficulties, providing them with tailored educational activities that reinforce their cognitive and social development. In addition to their adaptability, social robots play a crucial role in fostering collaboration and communication among students (34). Collaborative learning, where students work together to achieve shared educational goals, is essential to early childhood development (35). Robots can facilitate group activities and encourage students to communicate with each other, enhancing their social skills and teamwork abilities. The interactive nature of robots enables them to mediate between students, guide discussions, and ensure that all participants are actively engaged in the learning process (36). This improves students' academic outcomes and helps them develop essential life skills, such as problem-solving, critical thinking, and interpersonal communication.

Social robots are particularly effective in language acquisition, a crucial component of early childhood education (37). (11) has demonstrated that children who interact with robots as part of their language learning curriculum improve vocabulary, pronunciation, and overall language proficiency. In such environments, robots can facilitate language learning by offering consistent, patient, and personalised instruction, ensuring no student is left behind. The importance of social robots in enhancing engagement and motivation cannot be overstated. Robots provide an effective means of capturing and maintaining interest in early childhood and primary education, where attention spans are short, and students require constant stimulation. (38) demonstrated that students are more likely to remain engaged when interacting with robots, partly due to the novelty of the technology and partly due to the interactive learning experience they provide. Robots encourage students to actively participate in the learning process through games, quizzes, or problem-solving activities. This engagement is essential for promoting deeper learning and retention of information.

### 2.1.2 Robots in Language Learning

The integration of social robots into educational settings has shown significant promise, particularly in the area of language learning. Language acquisition, a critical aspect of early childhood education, requires tools and methods that cater to young learners' diverse cognitive and developmental needs (27). Social robots engage learners through multimodal approaches that combine auditory, visual, and kinesthetic elements (39). (35) This is crucial in early childhood, where young learners benefit from varied stimuli that reinforce learning. (40) demonstrated that children interacting with robots substantially improved vocabulary retention and sentence formation. This suggests that robots capture attention and enhance learning effectiveness by presenting information in ways that appeal to multiple senses. This multimodal experience is key to the cognitive development of young children in a critical language acquisition period.

The adaptability of social robots also plays a pivotal role in personalised education. In traditional classroom settings, the pace of language instruction may sometimes be aligned with the varying learning speeds of individual students (41). However, social robots can adjust their teaching approach based on a learner's proficiency level. This ensures that each student receives a personalised learning experience catering to their needs. For instance, robots can slow the pace of instruction for beginners or switch to more advanced content for more proficient learners, providing tailored feedback and encouragement. (42) this adaptive learning significantly improves engagement and retention among young learners, as the material remains challenging yet accessible. The capability of social robots to teach multiple languages makes them particularly valuable in multicultural and multilingual classrooms (43). In an increasingly globalised world, classrooms often consist of students from diverse linguistic backgrounds, and traditional language learning methods may struggle to accommodate these variations. Social robots, however, are equipped with multilingual functions, allowing them to switch between languages based on the student's background (44).

In the context of second-language learning, social robots have shown remarkable potential. Learning a second language can be challenging, particularly for young learners who may simultaneously develop proficiency in their first language. However, robots can assist by offering consistent practice in first and second languages. In multicultural classrooms, robots can be programmed to teach the primary curriculum in one language while offering supplementary lessons in the student's native tongues. This dual-language capability ensures that students do not fall behind in their education while learning a new language. A practical example of this is illustrated in the code provided in Scenario 1, where the robot greets children in their native language, creating an environment of inclusivity. This interaction highlights how robots can adapt to diverse classroom settings, promoting language learning and cultural sensitivity. In multilingual environments, such as those found in many urban schools, the ability of robots to bridge language gaps is invaluable (40). They ensure that no student is left behind, regardless of their linguistic background, and contribute to a more equitable learning experience for all.

Additionally, social robots provide an avenue for practising conversational skills, a crucial component of language learning. Unlike static language-learning tools such as textbooks or pre-recorded lessons, social robots can engage in real-time conversations with students (39). These interactions can simulate natural language use, allowing students to practice speaking and listening dynamically and interactively (27). Real-time conversation practice is especially beneficial in developing oral language skills, which are often more challenging to cultivate in traditional classroom settings (44). The immediate feedback provided by robots helps students refine their pronunciation, grammar, and conversational fluency, contributing to a more profound mastery of the language. Moreover, robots offer a non-judgmental platform for students to practice their language skills. In many cases, students learning a second language may feel anxious or self-conscious about making mistakes in front of their peers or teachers. Social robots provide a safe space where students can practice without fear of embarrassment or criticism. This aspect of social robots can be beneficial in promoting language learning among shy or introverted students, who may otherwise struggle to participate in traditional classroom discussions. Despite the many advantages of using robots for language learning, it is essential to acknowledge potential challenges. (40) a technical robot's ability to process and respond accurately to various dialects or accents can delay communication. Additionally, there is a risk that over-reliance on robots may detract from human interaction, which remains a fundamental component of language acquisition (35). However, when integrated thoughtfully into the broader educational framework, social robots can complement traditional teaching methods, enhancing language learning in previously unattainable ways.

### 2.1.3 Social Robots and Special Needs Education

Social robots have demonstrated significant potential in supporting the educational needs of students with special needs, particularly those with autism spectrum disorders (ASD) and other learning disabilities (37). These students often face challenges in traditional educational settings where verbal and written communication are predominant teaching methods. Such conventional methods may not be conducive to their learning styles, creating a need for alternative pedagogical tools that can cater to their specific needs (45). Social robots offer a solution by providing a non-threatening, interactive, and engaging way for these children to interact with educational content. The ability of robots to deliver personalised learning experiences, repeat tasks without fatigue, and provide immediate feedback makes them particularly suitable for students who benefit from a more individualised approach to education. (46) underscores the effectiveness of social robots in improving social communication skills among children with ASD. (47) note that one of the core challenges children with ASD face is difficulty in interpreting social cues, such as facial expressions, tone of voice, and body language. With their programmable features, social robots can be designed to use exaggerated facial expressions and body movements, making it easier for children with ASD to recognise and understand these cues (48). Furthermore, robots can engage in repetitive interactions without showing signs of frustration or impatience, offering a consistent learning experience that aligns with the needs of children requiring additional time and support to master social and communication skills.

In language learning, social robots have proven to be highly beneficial (49). For children with ASD, who often struggle with verbal communication, social robots can serve as both language models and conversational partners (50). Robots can help students practice vocabulary, sentence formation, and conversational turn-taking through structured interactions. For example, in Scenario 3 (Code 1), the robot actively engages children by asking them to describe objects they are playing with. This prompts the children to not only use descriptive language but also to participate in an interactive dialogue. Such engagement is crucial in helping children with special needs improve their language and communication abilities, which are often areas of difficulty for those with ASD (51). Moreover, social robots can raise shared activities, allowing children with special needs to develop collaborative skills in a structured, predictable environment. This is particularly important for students with ASD, who may find it challenging to engage in unstructured or spontaneous social interactions. Social robots, therefore, provide a controlled environment in which these students can practice social interactions in a repetitive, consistent manner, helping to build their confidence and competence over time.

Social robots' visual and auditory capabilities further enhance their effectiveness in supporting students with learning disabilities. Many children with ASD, for instance, benefit from visual learning aids, as they often find it easier to process visual information than verbal instructions (52). Social robots can be programmed to incorporate visual elements such as pictures, symbols, or animations into their lessons, making the learning experience more accessible and engaging for students who struggle with verbal communication. Additionally, the robots' ability to use auditory prompts and responses enables them to cater to students with varying learning preferences, ensuring a more inclusive approach to education (53). In addition to their educational benefits, social robots play a crucial role in reducing anxiety and creating a sense of comfort for students with special needs (54). Traditional classroom settings can sometimes be overwhelming for these students, mainly when they must interact with unfamiliar adults or peers. With their predictable and non-judgmental nature, social robots offer a safe space for students to engage in learning activities without the pressure of social scrutiny. This can significantly reduce anxiety levels in children with ASD, who may otherwise experience stress in more traditional social interactions. Another critical aspect of social robots in special needs education is their ability to adapt their behaviour based on individual student responses. Social robots can track students' progress in real-time, adjusting their teaching methods and difficulty levels to match the student's needs (55). This level of personalisation is precious in special education, where students often require tailored learning experiences that consider their unique strengths and challenges (56). By providing individualised support, social robots can ensure that each student receives the attention they need to succeed.

### 2.1.4 Effectiveness of Robots in Engaging Students

The effectiveness of robots in engaging students, particularly in early childhood and primary education, has been widely documented. Young children are naturally inquisitive and tend to be more willing to interact with novel technologies, making them ideal candidates for robot-assisted learning environments (57). Robots provide an interactive and dynamic platform for presenting educational content, thus transforming the traditional learning experience into a more engaging and enjoyable. Their capacity to offer immediate feedback and adapt their responses based on a student's input allows robots to surpass conventional digital tools such as tablets regarding responsiveness and personalisation. This real-time adaptability is critical in maintaining student engagement and improving learning outcomes.

One of the key advantages of using robots in education is their ability to create a more immersive and interactive learning environment (58). (59) (60) have shown that children are likelier to remain engaged when interacting with robots, as robots can simulate real-life situations and offer hands-on learning opportunities. For instance, in language learning, robots can serve as conversation partners, providing students with a simulated social environment to practice their communication skills (61). This is particularly beneficial in early childhood education, where language development is crucial to cognitive growth (62). In Scenario 2 (Code 2), the robot participates in symbolic games with the children, an excellent example of how robots can foster creativity and cognitive development. During these games, the robot asks questions about what the children are doing and even pretends to eat or drink the food the children offer. This form of interaction encourages symbolic play and fosters a sense of playfulness and imagination, both of which are vital for early childhood cognitive development (13). By participating in the children's play, the robot builds a rapport with the students, making them feel more at ease and engaged in the learning process. (63) indicates that children who feel emotionally connected to their learning tools are likelier to stay engaged and retain information.

Additionally, robots can cater to the diverse learning needs of students by offering personalised learning experiences. Many classrooms today are multicultural and inclusive, requiring teaching tools that can address the varied needs of students. Robots are particularly well-suited to this task, as they can adapt their interactions based on a student's learning style, pace, and preferences. For instance, a robot could slow down its speech for students learning a second language or use visual aids to assist children with learning disabilities (64). This flexibility is significant in inclusive classrooms, where students may have varying ability levels and need different support to succeed. The use of robots in engaging students also extends to children with special needs, such as those with ASD. (63) demonstrated that children with ASD often respond well to robot-assisted learning, as robots offer a predictable and non-judgmental social partner, which can help these children improve their social communication skills (56). In such cases, robots serve as an effective tool for engaging students who might otherwise struggle with traditional forms of education. The robot's ability to repeat instructions, provide consistent feedback, and adapt its behaviour based on the child's responses makes it an invaluable resource in special education settings (64).

Traditional teaching methods often involve a one-size-fits-all approach, which may not cater to the individual needs of each student. Robots, however, can deliver personalised instruction tailored to the student's specific learning needs. This level of personalisation helps to enhance student engagement, as the content is presented in a way that is both relevant and accessible to the learner (61). Robots can also facilitate collaborative learning, where students work together with the robot to solve problems or complete tasks, promoting teamwork and communication skills (57). In multicultural and inclusive educational settings, robots can help foster a more equitable learning environment. For example, robots can be programmed to greet students in their native language or provide culturally relevant examples during lessons, making learning more accessible for students from diverse backgrounds (58). This enhances student engagement and promotes inclusivity by ensuring that all students feel represented and valued in the classroom.

## 2.2 Robots for Intercultural Pedagogy

### 2.2.1 The Role of Robots in Fostering Cultural Awareness and Inclusion

In an increasingly globalised world, the role of education extends beyond the transmission of knowledge to the fostering of cultural awareness and inclusion (65). With their advanced capabilities, social robots are emerging as valuable tools for promoting intercultural pedagogy by acting as cultural mediators (27). These robots possess the ability to interact with students from diverse backgrounds, facilitate cultural exchange, and contribute to a more inclusive and supportive learning environment (66). Social robots can be highly effective in fostering cultural awareness due to their capacity for personalised interaction. Their ability to communicate in multiple languages and present culturally relevant content is particularly significant in multicultural classrooms (67). Robots can be programmed to greet students in their native language, share stories or traditions specific to different cultures, and encourage students to express aspects of their heritage (68). These activities enhance cultural awareness and foster a sense of belonging among students from diverse backgrounds, helping them feel valued and included (69). A significant aspect of cultural inclusion is recognising and validating a student's cultural identity. By engaging with students in a culturally sensitive manner, social robots can facilitate this validation. For instance, when a robot greets a student in their native language, it acknowledges the student's linguistic background and creates a welcoming atmosphere that promotes inclusivity. This approach aligns with Vygotsky's social development theory, which posits that learning is inherently a social process, and interactions with more knowledgeable others, such as a culturally responsive robot, can support cognitive and social development (69).

Regarding practical application, robots like those described in Scenario 3 (Code 2) encourage children to describe aspects of their culture, such as their clothes or games. This interactive approach fosters cultural exchange, allowing children to share their experiences while learning about others. In doing so, robots create an environment where diversity is celebrated and cultural awareness is deepened. The robot's questions about the children's culture stimulate discussions around diversity, helping students develop a sense of pride in their heritage while promoting mutual understanding and respect among their peers (70). The role of social robots in facilitating intercultural dialogue is further supported by projects like Culture-Aware Robots and Environmental Sensor Systems for Elderly Support (CARESSES), which developed culture-aware robots capable of adapting their behaviour based on the user's cultural background. Although primarily focused on elderly care, CARESSES provides important insights into how robots can promote cultural awareness in educational settings (71). By greeting users in their native language and adjusting their behaviour according to cultural norms, CARESSES robots have demonstrated the potential of social robotics to bridge cultural gaps and promote inclusion (27). Similarly, these capabilities can be leveraged in education to foster a sense of belonging in multicultural classrooms.

Additionally, teachers and educators must ensure that social robots do not alienate students who may be less familiar with technology. For some students, particularly those from disadvantaged backgrounds, interactions with robots may feel intimidating or uncomfortable. Educators must provide sufficient support to ensure that all students, regardless of their familiarity with technology, can benefit from the robot's cultural mediation capabilities (67). This may involve integrating robots gradually into the classroom, ensuring that students can become familiar with the technology in a supportive and inclusive manner. Technical limitations also pose a challenge when using social robots for intercultural pedagogy (72). (65) argue that robots can switch between languages, so their proficiency in lesser-known or underrepresented languages may be limited, which could create barriers in some multicultural classrooms. Addressing these limitations requires continuous development and refinement of robot language capabilities, ensuring they can communicate effectively with students from diverse linguistic backgrounds (68). This is particularly important in classrooms where multiple languages are spoken, as the robot's ability to switch seamlessly between languages can significantly impact its effectiveness as a cultural mediator. Despite these challenges, social robots have immense potential to foster cultural awareness and inclusion in education. By promoting intercultural dialogue, celebrating diversity, and facilitating personalised learning experiences, robots can contribute to a more inclusive and culturally responsive learning environment. Their ability to engage students in culturally relevant ways, combined with their potential to adapt to the unique needs of individual learners, makes them valuable tools for modern education.

### 2.2.2 Relevant Projects in Intercultural Robotics

Several projects have investigated the role of culture-aware robotics in educational settings to foster cultural awareness, promote inclusivity, and support intercultural pedagogy. These projects have demonstrated how robots, as social agents, can facilitate meaningful intercultural interactions and help bridge cultural gaps in diverse learning environments.

One such prominent project is the CARESSES, which has been instrumental in advancing socially intelligent robots capable of adapting to users' cultural backgrounds (73). While primarily aimed at elderly care, its educational implications are significant, especially in multicultural classrooms. The CARESSES project was designed to develop robots capable of culturally appropriate behaviour, including language, gestures, and communication styles (74). These robots can recognise the individual's cultural background and adjust their interactions accordingly, fostering more personalised and relevant exchanges (73). In educational settings, these robots can support students from diverse linguistic and cultural backgrounds by engaging with them in culturally appropriate ways (73). For example, in multicultural classrooms, CARESSES robots can greet students in their native language, enhancing students' sense of belonging and inclusion. Moreover, the ability of these robots to adjust their communication style and gestures according to cultural norms can help prevent misunderstandings and promote a more inclusive environment for students from various cultural backgrounds.

In addition to CARESSES, the Embodied-perceptive Tutors for Empathy-based Learning (EMOTE) project focuses on integrating robots into educational environments with a strong emphasis on empathy and emotional understanding (75). EMOTE robots are designed to adapt their emotional responses to the needs of students, providing a more empathetic learning experience. (76) highlights the potential of robots to facilitate learning and create emotional connections with students, which is particularly important in intercultural settings where empathy and understanding are key to fostering mutual respect and cooperation. EMOTE robots can help students from different cultures feel understood and supported by recognising and responding to their emotional states (76). For instance, in a classroom where students from different cultural backgrounds may struggle with language barriers or feelings of isolation, an EMOTE robot can detect frustration or disengagement and adapt its responses to provide encouragement and emotional support (75). This ability to recognise and respond to students' emotional needs is crucial in fostering a positive and inclusive learning environment where students feel valued and respected regardless of their cultural background.

The RoboLang project is another notable example focusing on language acquisition and intercultural communication (77). RoboLang aims to develop robots that can act as language tutors, supporting both language learning and cultural exchange (78). These robots are designed to engage students in interactive language-learning activities, using culturally relevant content to make learning more engaging and meaningful (79). For instance, RoboLang robots can present language lessons incorporating cultural references, such as traditional stories, customs, and holidays from the students' cultural backgrounds, thereby fostering a deeper connection between language learning and cultural awareness. RoboLang robots have shown particular promise in multilingual and multicultural classrooms, where students may learn a second language while navigating a new cultural environment. By incorporating cultural references into language lessons, these robots help students understand the language and the cultural context in which the language is used. This approach supports a more holistic form of language learning, where students gain linguistic and cultural competence, thus fostering greater intercultural understanding.

The Mitsubishi Electric Factory Automation (MELFA) Learning & Development (L&D) project also emphasises the role of robotics in promoting intercultural communication and inclusivity (80). MELFA robots are designed to facilitate collaborative learning activities, encouraging students from diverse cultural backgrounds to work together on group projects. These robots help bridge cultural divides by fostering teamwork and cooperation, allowing students to learn from one another's perspectives (81). The collaborative nature of MELFA robots promotes dialogue and interaction between students, helping to break down cultural barriers and build mutual understanding (82). MELFA robots can mediate group activities in a multicultural classroom, encouraging students from different cultures to share their ideas and experiences. For example, in a project where students are tasked with creating a presentation on different cultural traditions, the MELFA robot can guide them in discussing their respective cultures and identifying commonalities and differences (83). By facilitating these intercultural exchanges, MELFA robots help students better appreciate cultural diversity and foster a more inclusive learning environment.

The above analysis shows that CARESSES, EMOTE, RoboLang and MELFA ŁD have contributed to developing socially intelligent robots that promote intercultural awareness and inclusivity in educational settings. These robots can adapt their behaviour to the cultural background of students, facilitate language learning and cultural exchange, and foster emotional connections with students from diverse backgrounds. By creating a more culturally responsive and inclusive learning environment, social robots can potentially transform education, particularly in multicultural classrooms where cultural awareness and inclusion are paramount. Integrating these robots into educational settings also raises important ethical considerations, such as avoiding cultural stereotyping and ensuring that robots present a balanced and accurate representation of different cultures. However, the potential benefits of using culture-aware robots to foster intercultural dialogue and inclusion far outweigh these challenges, making them a valuable tool in modern education.

### 2.2.3 Ethical Considerations in Multicultural Educational Settings

Integrating social robots in multicultural educational settings presents several ethical concerns that must be addressed to ensure their effectiveness and inclusivity. One of the foremost concerns is the risk of reinforcing cultural stereotypes (84). Robots designed to present or engage with certain cultural content may inadvertently perpetuate a narrow or biased representation of that culture (85). As humans primarily program robots, there is a risk that they may convey an incomplete or stereotypical portrayal of a culture, especially when its diversity is not adequately represented (86). This concern is particularly significant in educational environments that foster cultural awareness and sensitivity. Social robots in multicultural classrooms must be programmed with diverse and accurate cultural content to avoid reinforcing stereotypes (87). Developing these robots must involve consultation with cultural studies and education experts to ensure that the information they provide is inclusive and representative of the various facets of the cultures they aim to depict. For instance, a robot designed to engage with students about cultural holidays should be able to discuss how a holiday might be celebrated within a given cultural group, ensuring that students receive a well-rounded and accurate understanding (88).

Furthermore, introducing social robots in multicultural settings must be done with sensitivity to students' diverse needs and preferences. While some students may be eager to interact with new technology, others might feel uncomfortable or even alienated by the presence of robots in the classroom, mainly if they come from backgrounds with limited access to technology. It is important that educators carefully consider how to introduce robots in a way that is inclusive and respectful of these differences (89). Teachers must ensure that no student feels marginalised by the introduction of robots, especially those who may not be as familiar with or comfortable using technology (11). In this regard, the role of the teacher is crucial in facilitating these interactions and ensuring that all students feel included and engaged in the learning process. Another significant ethical concern relates to the technical limitations of robots, particularly in their language proficiency (90). While many social robots are designed to switch between languages, their language capabilities are limited by the software they run on, which may not include support for less commonly spoken languages (91). This can lead to communication barriers in multicultural classrooms where students may speak various languages. If a robot is unable to communicate effectively in a student's native language, it may hinder the student's ability to engage with the robot and the educational material it presents (92). For instance, in Scenario 1, the robot greets children in their native language. However, if the robot's software does not support a particular language or its language proficiency is limited, the interaction may feel disjointed or even exclusionary for some students.

In addition to language barriers, social robots may face challenges in understanding and responding to cultural nuances. For example, in Scenario 3, the robot asks children to describe aspects of their culture. Therefore, it is essential that the robot can comprehend and appropriately respond to the cultural details shared by the students. If the robot fails to do so, it may give the impression that the cultural information is not valued or understood, which could undermine the robot's role in fostering intercultural dialogue (69). Therefore, it is crucial that social robots in multicultural educational settings are equipped with language capabilities and the ability to understand and engage with cultural content in a meaningful way (93). Despite these ethical concerns, the potential of social robots to facilitate intercultural pedagogy is significant. Social robots offer a unique opportunity to promote cultural exchange, facilitate language learning, and support social inclusion in diverse classrooms. By providing students with opportunities to interact with technology in a way that is tailored to their cultural background, robots can help bridge cultural divides and create a more inclusive learning environment (69). For instance, in Scenario 2, the robot engages with children in symbolic games, asking culturally relevant questions about the children's play activities. This type of interaction promotes cultural awareness and allows students to learn about each other's cultural backgrounds playfully and engagingly.

Moreover, social robots can act as cultural mediators, facilitating discussions about culture and diversity that might need help to navigate independently. By presenting culturally relevant content in a neutral and non-judgmental manner, robots can help foster an open and inclusive dialogue among students. This can be particularly valuable in classrooms with students from diverse cultural backgrounds, where discussions about culture and identity may be sensitive or challenging for both students and teachers. While the use of social robots in multicultural educational settings offers many benefits, it also raises important ethical considerations that must be addressed to ensure that these technologies are used in an inclusive and culturally sensitive manner. Educators and developers must work together to ensure that robots are programmed with accurate and diverse cultural information, that their introduction in classrooms is sensitive to the needs of all students, and that their technical limitations are managed to avoid communication barriers and cultural misunderstandings. Social robots can increase intercultural pedagogy and create more inclusive and engaging learning environments for these ethical concerns.

### 2.2.4 Summary

This provides a comprehensive review of social robotics in education, particularly focusing on primary and early childhood settings. It discusses the role of social robots in shifting traditional teaching to more interactive, student-centered methods, highlighting their effectiveness in language learning, individualized support, and engagement for students, including those with special needs. The chapter explores robots' role in intercultural pedagogy, showcasing their ability to foster cultural awareness and inclusion. It also reviews significant projects like CARESSES, RoboLang, and EMOTE. Ethical considerations, including the risks of cultural stereotyping and ensuring inclusive technological use, are also examined in this chapter.

# Chapter 3

# Methodology

## 3.1 Overview

The methodology chapter details the design, development, and implementation of Android-based applications for the Buddy robot to enhance multicultural inclusivity in educational settings. It covers the conceptualization of Buddy's interactions in three scenarios: multilingual engagement, cultural exploration, and cultural exchange. The chapter explains the use of the Buddy SDK, Microsoft Cognitive Services for speech recognition, and multimedia integration. It outlines the technical aspects, including speech synthesis, gesture simulation, and robot movement. The chapter also describes how Buddy's interactions were tailored to meet diverse classroom needs, focusing on personalization, inclusivity, and adaptive learning for children from various cultural backgrounds.

## 3.2 Robot Description

Buddy, developed by Blue Frog Robotics, is an advanced social robot designed for educational and social interaction purposes. Standing at a height suitable for children, Buddy's appearance is both approachable and friendly, with a humanoid form that includes a digital face capable of expressing a wide range of emotions. Its facial features, such as eyes and mouth, change dynamically based on interactions, allowing it to convey emotions like happiness, curiosity, and empathy. Buddy is equipped with a mobile base, which allows it to move around smoothly, making it more interactive and engaging in various settings such as classrooms or at home.

Technologically, Buddy incorporates a range of sensors, actuators, and AI-driven capabilities to interact seamlessly with humans. These include tactile sensors located on its head and body, allowing Buddy to respond to touch and interact with users based on physical contact. The robot also comes with advanced visual and audio recognition systems. It has proximity sensors to detect objects and avoid obstacles, as well as a built-in camera that enables face recognition and tracking. This ensures that Buddy can follow movements and recognize individuals in a room. Additionally, its microphone array helps Buddy localize sounds, recognizing where voices are coming from and allowing for seamless interaction. Buddy's ability to engage in verbal communication is facilitated by sophisticated speech-to-text and text-to-speech functionalities. Using ReadSpeaker API for text-to-speech (TTS) and Cerence for speech-to-text (STT), Buddy can comprehend and produce speech in multiple languages. This multilingual capability allows Buddy to communicate in English, French, Italian, and other languages, making it particularly useful in multicultural environments.

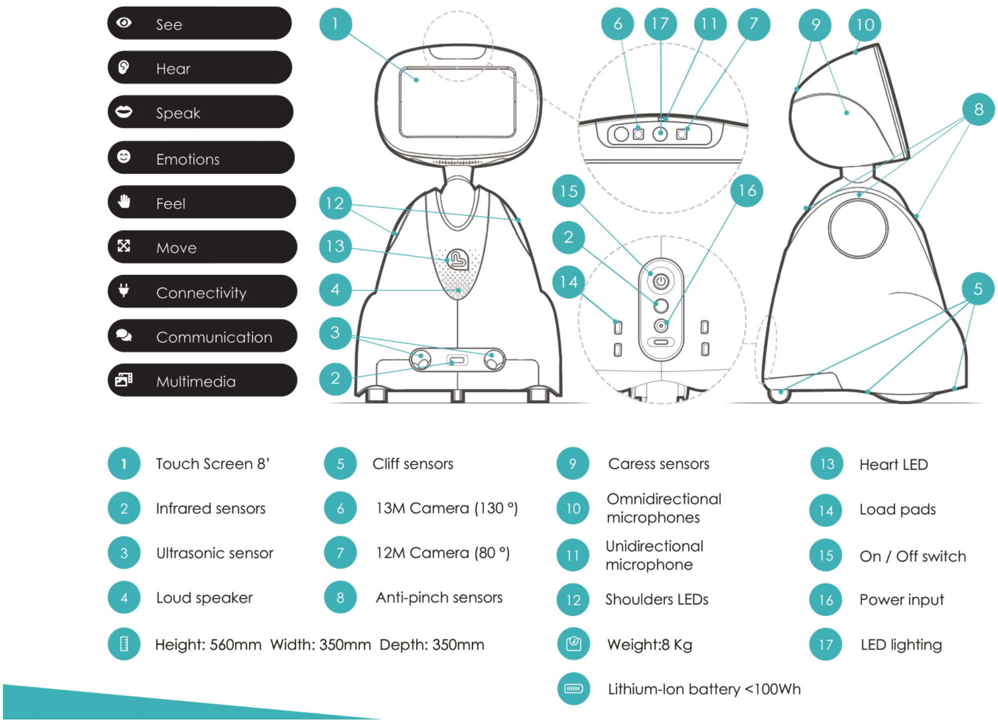


Figure 3.1: Capacities and Technical Specification of Buddy Robot

Another notable aspect of Buddy is its capacity for movement. Buddy's mobility is powered by wheels, enabling it to navigate its environment. The wheels are controlled via precise motor actuators, and the robot can rotate, move forward or backward, and adjust its speed depending on the interaction. It can even perform basic head movements, such as nodding to indicate agreement or shaking to show disagreement. These motorized actions make Buddy's interactions more lifelike and relatable to children, fostering engagement and interaction in educational settings. Buddy's design extends to its lighting system as well. The robot is equipped with LED lights in its shoulders and heart area, which can change colors and blink in response to different stimuli or commands. For example, Buddy can use its LEDs to visually represent its mood, with specific colors corresponding to various emotional states. This enhances the sensory interaction between Buddy and its users, providing visual cues that complement its verbal communication and facial expressions. Therefore, Buddy is a versatile and engaging educational robot designed to promote interaction, learning, and emotional connection. Its combination of movement, facial expressions, touch sensors, and vocal abilities creates a dynamic platform for children to interact with, making it a valuable tool for enhancing learning, socialization, and cultural exchange in diverse educational contexts.

## 3.3 Speech Recognition and Language Integration

Microsoft Azure's speech recognition system support multiple languages, tailored explicitly for recognising children's voices and generating language-specific responses (94). Microsoft Azure's Cognitive Services offer a robust framework for speech recognition, which serves as the backbone of the language integration feature in social robots. (95) Azure's speech services are designed to convert spoken language into text, recognise different languages, and generate corresponding responses in the appropriate language (96). In an educational setting, this system enables social robots to interact with students in their native language or a second language they are learning, enhancing their educational experience (97).

Microsoft Azure currently supports over 100 languages and dialects, making it an ideal tool for multilingual classrooms where students speak various languages. Moreover, Azure's language models are constantly updated to improve accuracy and add support for additional languages, ensuring that the system remains relevant in increasingly diverse educational environments (98). Azure's speech recognition system is cloud-based, requiring minimal hardware resources from the robots (99). Instead, the robots are equipped with microphones and connected to Azure's cloud services, where speech processing and language recognition occur. The system captures the audio input, sends it to the cloud for analysis, and returns the transcribed text and corresponding response to the robot in real-time (98). This process is efficient and scalable, allowing robots to operate seamlessly in various educational settings without extensive computational power.

## 3.4 Robot-Child Interaction Design

The design of social robots for interaction with children in educational environments is pivotal for developing engagement, inclusivity, and cultural awareness. In this project, Buddy, a humanoid social robot, interacts with children through gestures, verbal communication, and multimedia aids. Although Buddy does not possess physical hands, using images and animations to simulate gestures is a key part of its interaction design. In the context of inclusivity, the robot displays a waving hand image on its tablet to greet children or say goodbye, helping establish a welcoming and friendly atmosphere. Images, rather than physical gestures, ensure that Buddy can still participate in social rituals like waving, which is universally recognised and understood across different cultures. In scenario 1, for example, Buddy sits in a circle with children and, when greeted by the educator or the children, it responds by displaying a waving hand image on its screen. This visually stimulating interaction makes the robot approachable, bridging the gap between the human and robotic experience. The image of a waving hand is a subtle but effective way to replicate a culturally neutral gesture, reinforcing the robot's ability to connect with children from diverse backgrounds.

Similarly, Buddy can display images representing different stages of an activity or song, which helps children follow along with the lesson or group activity. Buddy's ability to show culturally relevant images, such as pictures of traditional foods, clothing, or games from different countries, also enhances its effectiveness in multicultural classrooms. In scenario 2, Buddy displays images of food and drinks from other cultures, such as tea, to introduce students to diverse cultural practices. This visual aid not only promotes inclusivity but also helps children who may be visual learners or who may not be familiar with the languages being spoken. By displaying these images, Buddy ensures that children can engage with the robot and the lesson meaningfully, regardless of their language or cultural background. Through its voice, Buddy invites interaction with the children by asking questions like, "Can I play with you?" This verbal prompt encourages children to include Buddy in activities and develops a collaborative environment. When the children offer Buddy food or drinks as part of the symbolic play, Buddy verbally thanks them and pretends to consume the food, enhancing the realism of the interaction by imitating the sound of drinking and adding phrases like, "This is very good!" These verbal interactions create a playful and imaginative atmosphere where the children can engage with Buddy as if he were one of their peers.

In Scenario 3, when Buddy invites children to participate in a game, it uses expressions to signal different emotions, such as surprise or joy, depending on how the children respond. For example, when children correctly guess the name of an object or image displayed on the robot's tablet, Buddy may display an excited or happy expression to reinforce positive behaviour and motivate further participation. This dynamic use of expressions creates a more immersive and emotionally engaging interaction, which is particularly important for young children who rely heavily on non-verbal cues in social communication. This raises cultural awareness and encourages the children to share aspects of their own cultures. Buddy's questions, such as "Can you tell me what games you play?" or "What kind of clothes do you wear?" stimulate the children to reflect on their cultural identities and share them with their peers, promoting inclusivity and mutual respect. The robot reinforces cultural exchange and language learning by inviting children to describe their cultural practices and teaching Buddy new words related to the objects they are playing with. In one instance, Buddy may ask a child to teach it the name of a toy or accessory in their language, thereby validating the child's cultural knowledge and empowering them as contributors to the learning process. This reciprocal learning dynamic strengthens Buddy's role as an inclusive and culturally responsive educational tool.

## 3.5 Overview of Applications

The use of robotics in education has emerged as a transformative approach to enhancing classroom learning, particularly through the integration of social robots like Buddy. The development of applications for Buddy enables interactive, culturally responsive, and engaging learning experiences for children. These applications, centered on the themes of inclusivity, multicultural interaction, and language learning, offer a dynamic platform for social interaction in educational settings. By responding to greetings, participating in symbolic play, and fostering cultural exchanges, Buddy serves as a bridge between technology and education, enhancing students' cognitive and social development.

The first application focuses on integrating Buddy into classroom settings during circle time, an essential part of early childhood education that promotes social interaction and group learning. The application enables Buddy to join children during circle time and respond to their greetings in multiple languages, making the robot a versatile tool for fostering inclusivity and multicultural interaction. By simulating human gestures, such as waving goodbye or smiling, Buddy engages children in an interactive manner. Additionally, the robot's multimedia capabilities, such as displaying a waving hand or pictograms for different greetings on its tablet, further enhance the learning environment. The key feature of this application is Buddy's ability to support multilingual interactions, which are critical in culturally diverse classrooms. The application uses speech recognition technology to detect and understand the language spoken by the child. Once the language is recognised, Buddy responds in the appropriate language using speech synthesis, offering personalised interactions that make children feel acknowledged and included, regardless of their cultural background. This capability is particularly useful in classrooms where children speak different languages, enabling teachers to create an inclusive learning environment. Another important aspect of this application is the display of gestures through multimedia aids. As Buddy does not have physical limbs, it compensates by showing a waving hand on its screen when greeting or bidding farewell. This gesture, while virtual, is perceived by children as a real interaction, fostering a sense of connection and enhancing engagement during classroom activities. The display of pictograms for non-verbal greetings also supports children with different learning needs, ensuring that everyone in the classroom can participate in the interactions. Technically, this application relies on a combination of speech recognition, speech synthesis, and multimedia display functionalities. The speech recognition system is designed to interpret children's spoken greetings in real-time, allowing Buddy to identify the language and respond appropriately. The speech synthesis module generates responses in the correct language, while the multimedia display supports the visual representation of gestures. Additionally, facial expression cues on Buddy's screen, such as smiling, are used to enhance emotional engagement with the children. These technological components work in tandem to create an immersive and interactive learning experience.

The second application builds upon the foundation of Buddy's interactive capabilities by extending its role into symbolic play and cultural exchange. Symbolic play, a critical aspect of child development, allows children to express themselves and explore different cultural identities through play activities. In this application, Buddy moves with the children to the symbolic game room, where it takes part in the game by asking children questions, accepting invitations to play, and engaging in discussions about food and customs from different cultures. This application is particularly designed to promote intercultural pedagogy by encouraging children to share their cultural knowledge and learn from one another. The integration of multilingual capabilities plays a central role in this application. Buddy can detect when children are speaking a language other than Italian and respond accordingly. This feature allows Buddy to participate in discussions about foods, customs, and traditions from various cultures, providing an opportunity for children to learn about diversity through play. For example, when children offer Buddy food or drinks, it responds by thanking them and pretending to eat or drink, accompanied by sound effects. It then introduces foods from "its" culture, creating an imaginative scenario where Buddy represents a robot from another planet. The robot can also display images of drinks from different cultures, such as tea, on its tablet, stimulating discussions about cultural differences. Furthermore, Buddy uses its language detection and translation capabilities to facilitate cultural exchanges. If children speak a language unfamiliar to others, Buddy can translate their words, fostering communication between children from different linguistic backgrounds. This application not only supports language learning but also promotes cultural awareness and inclusion, key goals of intercultural pedagogy. Buddy's role in symbolic play enables children to explore their identities while learning about the customs and traditions of others in a fun and engaging way. From a technical perspective, this application integrates gesture simulation, speech synthesis, and cultural content display to create an immersive play experience. Buddy's interactions are guided by speech recognition, allowing it to ask questions, respond to offers, and engage in cultural discussions. The use of emotion simulation through facial expressions, such as smiling when being offered food, adds a layer of realism to the interactions, making Buddy an engaging participant in the children's play. The robot's tablet plays a crucial role in displaying cultural content, supporting discussions about different foods and traditions. By combining these technical elements, Buddy enhances the symbolic play experience while promoting cultural understanding.

The third application is focused on cultural storytelling and knowledge sharing, with Buddy acting as a facilitator of discussions about different cultures. This application allows Buddy to move with children to the symbolic game room and initiate conversations about cultural topics such as traditional clothing, games, and customs. By asking direct questions and encouraging children to describe typical aspects of their cultures, Buddy creates an environment where cultural knowledge is shared in an interactive and engaging manner. In this application, Buddy's role extends beyond simply responding to greetings and participating in play. The robot actively stimulates conversations by asking children to teach it new words related to the objects they are playing with, such as dolls, clothes, or accessories. Additionally, Buddy organises games where children can guess the names of cultural items displayed on the tablet, fostering both language learning and cultural awareness. The robot's ability to translate words into different languages further supports these knowledge-sharing activities, allowing children from different backgrounds to communicate and participate fully in the discussions. The technical infrastructure for this application leverages speech recognition, gesture simulation, and multimedia display to facilitate cultural storytelling. The speech recognition system allows Buddy to listen to children's descriptions of cultural objects and generate appropriate responses using speech synthesis. The robot's tablet is used to display images of cultural items, such as traditional clothing, that children can guess during games. Buddy's ability to translate words into multiple languages is supported by advanced speech synthesis capabilities, ensuring that children from various linguistic backgrounds can contribute to the knowledge-sharing process.

## 3.6 Libraries used in Applications

In developing interactive applications for social robots such as Buddy, the choice of libraries plays a critical role in determining the functionality and user experience. Several key libraries are used in the application to enable speech recognition, gesture simulation, multimedia support, and permission management. This integration of various libraries is essential for creating a seamless and engaging interaction between Buddy and the children in educational settings.

The application makes use of Microsoft Cognitive Services for speech recognition and synthesis:

```java
import com.microsoft.cognitiveservices.speech.*;
```

Figure 3.2: Microsoft Azure Speech Recognition

One of the core libraries utilised in the application is the Speech SDK from Microsoft Cognitive Services, which is responsible for both speech recognition and speech synthesis. The Speech SDK allows Buddy to understand spoken commands and respond in natural language, which is central to the robot's interactive capabilities. Through the use of classes such as ‘SpeechConfig’, ‘SpeechRecognizer’, and ‘SpeechSynthesizer’, the robot can detect and interpret verbal inputs in multiple languages, facilitating an inclusive experience. The SpeechConfig class is initialized with a subscription key and service region, which links the application to Microsoft’s cloud-based speech services. The ‘SpeechRecognizer’ class processes real-time audio input from Buddy’s microphone, converting speech into text. This recognised text is then processed by the application logic to trigger appropriate responses, such as displaying gestures or playing multimedia content. Moreover, speech synthesis is implemented using the ‘SpeechSynthesizer’ class, allowing Buddy to speak responses generated by the application. The speech synthesis feature is particularly useful in multilingual environments, where Buddy can greet children in different languages, responding based on the detected language of the speaker. This powerful combination of speech recognition and synthesis enhances the robot’s interactivity, making it an effective tool for education and communication in diverse settings.

To control the Buddy robot's hardware and software capabilities, the application uses several imports from the Buddy SDK:

```
import com.bfr.buddy.ui.shared.FacialExpression;
import com.bfr.buddy.speech.shared.ISTTCallback;
import com.bfr.buddy.speech.shared.STTResultsData;
import com.bfr.buddy.usb.shared.IUsbCommadRsp;
import com.bfr.buddysdk.BuddyActivity;
import com.bfr.buddysdk.BuddySDK;
import com.bfr.buddysdk.services.speech.STTTask;
```

Figure 3.3: Buddy SDK's Libraries

Another crucial library integrated into the application is the BuddySDK, which provides the core interfaces for controlling Buddy's physical and virtual behaviors. The BuddyActivity class serves as the foundation for creating activities within Buddy's environment, offering a streamlined way to manage the robot's behavior in response to user inputs. This SDK also includes the 'BuddySDK.USB' interface, which allows the robot to execute physical movements such as rotating or advancing. These movements are essential for making Buddy appear more lifelike and engaging, particularly during interactive sessions with children. Additionally, the expression system is managed using the 'FacialExpression' class, which allows the application to control Buddy's screen-based facial expressions. By simulating moods and emotions through these facial expressions, Buddy can convey happiness, neutrality, or other emotional states, depending on the context of the interaction. This feature is particularly useful in scenarios where non-verbal communication is important for engaging children, helping them feel connected to the robot.

For efficient loading and handling of images, GIFs, and video files, the Glide library is imported:

```
import com.bumptech.glide.Glide;
import com.bumptech.glide.annotation.GlideModule;
import com.bumptech.glide.module.AppGlideModule;
```

Figure 3.4: Glide Libraries

The Glide library, a popular choice for handling multimedia content, is another key component of the application. Glide is used extensively for image loading and managing multimedia assets, such as videos and GIFs, on Buddy's screen. In situations where Buddy needs to simulate gestures, such as waving goodbye, the application can use Glide to load and display a GIF of a waving hand. This visual representation of gestures compensates for the robot's lack of physical arms, allowing Buddy to engage with children through virtual gestures. Glide's efficient image handling capabilities ensure that multimedia content is loaded quickly and smoothly, which is critical for maintaining the fluidity of interactions. The library's ability to handle both static and dynamic content—such as images, GIFs, and videos—provides versatility in how Buddy can interact with its users. Additionally, Glide's integration with Android's 'ImageView' ensures that the multimedia content is displayed at high quality, further enhancing the overall user experience.

The application uses several Android core libraries that are essential for permission management, multimedia handling, and intent launching. These imports include:

```
import android.Manifest;
import android.content.Context;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.graphics.Bitmap;
import android.net.Uri;
import android.os.Build;
import android.os.Bundle;
import android.os.Handler;
import android.os.Looper;
import android.provider.MediaStore;
import androidx.core.app.ActivityCompat;
import androidx.core.content.ContextCompat;
import androidx.core.content.FileProvider;
```

Figure 3.5: Android Core Libraries

In terms of permission management and core Android functionality, the application leverages several classes from the Android Core Libraries. One of the most important classes is ContextCompat, which is used to manage permissions at runtime. In modern Android development, explicit user permission is required for accessing sensitive data or hardware components, such as the microphone and external storage. The application checks whether the necessary permissions (such as audio recording and storage access) have been granted using 'ContextCompat.checkSelfPermission'. If the permissions are not granted, the application requests them from the user using 'ActivityCompat.requestPermissions'. This permission handling ensures that the application complies with Android's security model, protecting user privacy while enabling the necessary functionalities for speech recognition and multimedia display. Another important core library feature is Intent, a class used for launching activities within the Android ecosystem. Intents are used in this application to start activities for playing videos or displaying specific content on Buddy's screen. For instance, when the application needs to play a video in response to a recognised speech command, it uses an Intent to launch a video player activity, passing the video file URI as data. The Intent class simplifies the process of launching new activities and handling data within the application, making it an essential tool for multimedia management.

Furthermore, the application uses FileProvider for secure access to file URIs, particularly when dealing with video files stored on the device. Since Android restricts direct access to file paths for security reasons, FileProvider acts as an intermediary, granting the application permission to share files with other apps (such as a video player) without exposing sensitive file paths. This is crucial for enabling the smooth playback of video content, as it ensures that the necessary file permissions are handled securely and efficiently.

## 3.7 Technical Breakdown of Applications

The code for the Buddy robot application integrates multiple functionalities, including speech recognition, speech synthesis, robot movements, and multimedia interaction, to create a socially interactive experience for users, particularly children in educational settings. This application is built using a combination of Android's core libraries, the Buddy SDK, and Microsoft Cognitive Services, among others. In this detailed analysis, I will explain the code in technical depth, discussing the significance of various components, libraries, methods, and how they work together to create a seamless interactive experience.

```
package com.bfr.scenario1;
```

Figure 3.6: Package Name

The package structure com.bfr.scenario1 is used to group classes related to a specific scenario where Buddy interacts with users. Organising the application into packages helps in code modularisation and reuse, making the code more manageable, especially in large projects. In this case, the scenario1 package indicates that this code handles a specific scenario or use case for the robot, which involves interactions like greetings, media display, and robot movement.

```
public class MainActivity extends BuddyActivity
```

Figure 3.7: Main Activity

The statement ”public class MainActivity extends BuddyActivity” defines the fundamental structure of the application. In this context, MainActivity is the primary entry point of the app, where the program logic, user interactions, and core functionalities are managed. The declaration establishes MainActivity as a subclass of BuddyActivity, inheriting all of its functionalities. This enables seamless integration with the Buddy robot, a humanoid robot equipped with sensors, speech recognition, and multimedia display capabilities. By making MainActivity a public class, it becomes accessible across the entire application, allowing other components to interact with it when necessary. The inclusion of the ”public” keyword signifies that this class is available throughout the app’s context, making it integral to the application’s overall flow. Within the context of an Android application, activities act as the central unit of execution. Each activity represents a single user interface screen, and in this case, MainActivity acts as the main controller for interactions with Buddy. Through this class, the robot’s speech recognition, movement, and multimedia responses are coordinated, providing an interactive experience for the user.

The ”extends BuddyActivity” part of the declaration highlights that MainActivity is built on top of the existing functionality provided by BuddyActivity. This is crucial because BuddyActivity is a specialised class that encapsulates the methods and behaviors necessary to control Buddy. By extending it, MainActivity inherits all of the built-in methods that facilitate Buddy’s interactions, such as moving the robot, manipulating facial expressions, and handling speech recognition and synthesis. This inheritance simplifies the development process because it abstracts away the lower-level complexities of interacting directly with the robot’s hardware. Instead of managing hardware operations manually, the developer can focus on higher-level logic by using pre-defined methods in BuddyActivity.

The BuddyActivity class itself is a customised extension of Android’s AppCompatActivity. AppCompatActivity provides compatibility with different versions of Android, ensuring that the application can run smoothly across a range of devices, even those with older versions of the operating system. This inheritance is critical because it allows MainActivity to utilise Android’s lifecycle management, which includes handling activity states such as when the app starts, pauses, or resumes. The inheritance chain from AppCompatActivity to BuddyActivity, and finally to MainActivity, ensures that the app maintains both the standard Android behaviors and the specific functionalities needed to operate the Buddy robot effectively.

```
@Override  new *
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    recyclerViewFiles = findViewById(R.id.recyclerView_files);
    recyclerViewFiles.setLayoutManager(new GridLayoutManager( context: this,  spanCount: 3));
    videoAdapter = new VideoAdapter( context: this, videoNames, this::playVideo);  // Initialize adapter
    recyclerViewFiles.setAdapter(videoAdapter);  // Set adapter to RecyclerView

    initViews();
    //configureListeners();
    checkPermissions();
}
```

Figure 3.8: onCreate

The onCreate method is a crucial part of the Android activity lifecycle, responsible for initialising the activity and setting up the user interface and essential functionalities. In this case, the @Override annotation signifies that this method is overriding the onCreate method from the superclass, which is typically AppCompatActivity. The protected access modifier ensures that the method is accessible within its package and to subclasses. This method is called when the activity is first created, and it is responsible for setting up the initial state of the activity. The method begins by calling super.onCreate(savedInstanceState);, which invokes the parent class's onCreate method. This is essential because the parent class handles the fundamental setup required for the activity, such as managing the activity's lifecycle, initialising the components provided by Android, and restoring the state of the activity when it is recreated after being destroyed (for example, due to a configuration change). Skipping this call would prevent the proper initialisation of the activity and could cause issues in its functionality.

The next line, setContentView(R.layout.activity_main);, is used to set the layout for this activity. This method inflates the XML layout file activity_main, which defines the user interface components, such as buttons, text views, and images, that will be displayed to the user. This method binds the UI elements defined in the XML file to the activity so that they can be referenced and manipulated programmatically within the MainActivity class. By specifying the layout resource file, the Android framework knows how to render the UI on the screen. The code proceeds to initialise a RecyclerView component, which is a flexible and efficient way to display a large number of items in a scrollable list. This method returns a reference to the RecyclerView object, allowing the program to interact with it programmatically. The RecyclerView is a powerful widget used to display lists or grids of items, and it is commonly used in Android applications for content that needs to be presented dynamically. After finding the RecyclerView, the layout manager is set for the view using recyclerViewFiles.setLayoutManager(new GridLayoutManager(this, 3));. This line assigns a GridLayoutManager to the RecyclerView, specifying that the items will be displayed in a grid format with three columns. The layout manager is responsible for positioning the items within the RecyclerView and managing the scrolling behavior. In this case, using a grid layout is ideal for displaying visual content like video thumbnails, as it provides a clear and organised view of multiple items at once.

Next, the videoAdapter is initialised with the statement videoAdapter = new VideoAdapter(this, videoNames, this::playVideo);. The VideoAdapter is a custom adapter that binds the data (in this case, a list of video file names) to the views that are displayed within the RecyclerView. The adapter is essential for managing how data is displayed in each item of the list or grid. In this case, the adapter is passed three arguments: the context (this), the list of video names (videoNames), and a method reference (this::playVideo). The method reference allows the adapter to invoke the playVideo method whenever a video item is selected. This design ensures that the behavior for playing videos is modular and can be handled within the activity itself. Once the adapter is initialised, it is set to the RecyclerView through the statement recyclerViewFiles.setAdapter(videoAdapter);. This binds the adapter to the RecyclerView, allowing the adapter to manage how each video file is displayed and how user interactions with each item are handled. The adapter plays a crucial role in ensuring that the data is dynamically loaded and displayed as the user scrolls through the list or grid of items.

The initViews() method is then called, which likely contains additional setup for other user interface components within the activity includes initialisation of buttons, text views, and other views that the user may interact with. By separating this logic into its own method, the code remains clean and modular, adhering to best practices in software design. Lastly, the check Permissions () method is invoked. This method is responsible for checking whether the app has the necessary permissions to access certain system resources, such as external storage for reading video files. Android's permission system requires apps to explicitly request permission from the user for accessing sensitive resources. The checkPermissions() method likely ensures that the necessary permissions, such as accessing the device's external storage or microphone, are granted before proceeding with any functionality that requires them. If the permissions are not granted, the method may request them from the user, ensuring that the app can function correctly.

```java
private void checkPermissions() { 1 usage new *
    if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.M) {
        if (ContextCompat.checkSelfPermission( context: this, Manifest.permission.RECORD_AUDIO) != PackageManager.PERMISSION_GRANTED ||
                ContextCompat.checkSelfPermission( context: this, Manifest.permission.READ_EXTERNAL_STORAGE) != PackageManager.PERMISSION_GRANTED)

            ActivityCompat.requestPermissions( activity: this,
                    new String[]{Manifest.permission.RECORD_AUDIO, Manifest.permission.READ_EXTERNAL_STORAGE},
                    PERMISSIONS_REQUEST_CODE);
        } else {
            initializeRecognizer();
        }
    } else {
        initializeRecognizer();
    }
}
```

Figure 3.9: checkPermissions

The checkPermissions() method is responsible for ensuring that the application has the necessary permissions to perform key functions, such as recording audio for speech recognition and reading external storage for accessing media files. In the context of an Android application, managing permissions is crucial, especially with the introduction of dynamic permissions in Android 6.0 (API level 23) and higher. The method begins by checking the device's Android version using the Build.VERSION.SDK_INT comparison with Build.VERSION_CODES.M. This check is necessary because devices running Android versions before 6.0 manage permissions differently, where permissions are granted at the time of installation rather than during runtime. Therefore, the logic within this method ensures compatibility with both newer and older Android devices.

If the device is running Android 6.0 or above, the method proceeds to check whether the necessary permissions—Manifest.permission.RECORD_AUDIO for audio recording and Manifest.permission.READ_EXTERNAL_STORAGE for accessing external storage—have been granted. This is accomplished using the ContextCompat.checkSelfPermission() method, which returns whether the app has the specified permissions. If either of the permissions has not been granted, the method triggers a runtime request for the necessary permissions using the ActivityCompat.requestPermissions() function. This function takes three arguments: the context (this), an array of the required permissions, and a unique request code (PERMISSIONS_REQUEST_CODE). By requesting permissions dynamically at runtime, the application complies with the modern Android permission model, enhancing both security and user control. This step is essential because, without explicit user consent, the app cannot access sensitive system resources like the microphone or external storage.

If the necessary permissions are already granted, the method proceeds to call initializeRecognizer(), which sets up the speech recognition service. This ensures that the application can immediately start performing speech recognition tasks without any further delay. Similarly, if the device is running an Android version older than 6.0, the permissions would have already been granted at installation, and the method directly calls initializeRecognizer() without checking for runtime permissions. This compatibility with older Android versions ensures that the app functions seamlessly across a wide range of devices.

```java
@Override
public void onRequestPermissionsResult(int requestCode, @NonNull String[] permissions, @NonNull int[] grantResults) {
    super.onRequestPermissionsResult(requestCode, permissions, grantResults);
    if (requestCode == PERMISSIONS_REQUEST_CODE) {
        boolean allPermissionsGranted = true;
        for (int result : grantResults) {
            if (result != PackageManager.PERMISSION_GRANTED) {
                allPermissionsGranted = false;
                break;
            }
        }
        if (allPermissionsGranted) {
            Log.i(TAG, "onRequestPermissionsResult: All permissions granted. Initializing recognizer and loading videos.");
            initializeRecognizer();
            loadVideoFiles();  // Load video files once permissions are granted
        } else {
            Log.i(TAG, "onRequestPermissionsResult: Permissions not granted. Showing error message.");
            Toast.makeText(this, "Permissions are required for this app to function", Toast.LENGTH_SHORT).show();
        }
    }
}
```

Figure 3.10: onRequestPermissionResult

The onRequestPermissionsResult method is an essential override in Android's permission-handling process, serving as a callback for when the user responds to a permission request. This method is triggered after the app requests permissions at runtime, and it processes the user's decision on whether to grant or deny the requested permissions. The method is overridden to provide custom behavior depending on the outcome of the permission request, and it takes three arguments: requestCode, permissions, and grantResults. The first step in the method is to check whether the requestCode matches the PERMISSIONS_REQUEST_CODE used earlier in the checkPermissions() method. The requestCode ensures that the app is responding to the correct permission request, especially in cases where the app might request multiple sets of permissions for different features. In this scenario, the code handles the request for both RECORD_AUDIO and READ_EXTERNAL_STORAG permissions, which are essential for the app to record audio for speech recognition and access media files. Once the correct requestCode is confirmed, the method initialises a boolean flag, allPermissionsGranted, assuming initially that all requested permissions have been granted. It then iterates over the grantResults array, which contains the user's response for each requested permission. The loop checks if any permission in the grantResults array was denied (i.e., if the result is not PackageManager.PERMISSION_GRANTED). If any permission is denied, the allPermissionsGranted flag is set to false, and the loop breaks. This ensures that the app can handle situations where the user grants some permissions but denies others.

If all permissions are granted, the method proceeds with a log statement (Log.i(TAG, "onRequestPermissionsResult: All permissions granted.")) to indicate success and calls two critical methods: initializeRecognizer() and loadVideoFiles(). The former sets up the speech recognition service, while the latter loads video files into the app once the READ_EXTERNAL_STORAGE permission is granted. These method calls ensure that the app can start performing speech recognition tasks and access media content without further user interaction. If any permission is denied, the method logs a message to indicate that the necessary permissions were not granted, and displays a Toast message to the user, explaining that the app requires permissions to function properly. The Toast.makeText() method creates a small pop-up notification to inform the user that the app's functionality will be limited due to the lack of necessary permissions. This ensures that the user is aware of why certain features may not work as expected, providing transparency and clarity. By handling both permission-granting and denial scenarios effectively, this method plays a crucial role in ensuring that the app operates securely while respecting user consent.

```
private void initializeRecognizer() {  4 usages  new *
    SpeechConfig config = SpeechConfig.fromSubscription(SUBSCRIPTION_KEY, SERVICE_REGION);
    List<String> languages = Arrays.asList("en-US", "fr-FR", "it-IT");
    AutoDetectSourceLanguageConfig autoDetectSourceLanguageConfig = AutoDetectSourceLanguageConfig.fromLanguages(languages);

    recognizer = new SpeechRecognizer(config, autoDetectSourceLanguageConfig);

    recognizer.recognized.addEventListener((s, e) -> {
        if (e.getResult().getReason() == ResultReason.RecognizedSpeech) {
            Log.i(TAG, msg: "Recognized: " + e.getResult().getText());


            runOnUiThread(() -> recognizedText.setText("Recognized: " + e.getResult().getText()));
            String speechText = e.getResult().getText().toLowerCase();


            runOnUiThread(() -> {
                processSpeechCommand(speechText);
                updateImageViewBasedOnSpeech(speechText);
            });
        } else if (e.getResult().getReason() == ResultReason.NoMatch) {
            Log.i(TAG, msg: "No speech could be recognized.");
            runOnUiThread(() -> recognizedText.setText("No speech could be recognized."));
        }
    });
```

Figure 3.11: initializeRecognizer

The initializeRecognizer() method is responsible for setting up the speech recognition functionality using Microsoft's Cognitive Services Speech SDK. This method begins by creating a SpeechConfig object using the subscription key (SUBSCRIPTION_KEY) and the service region (SERVICE_REGION). The SpeechConfig is necessary for authenticating and configuring the connection to Microsoft's speech recognition service, allowing the application to process voice commands. Following the initialisation of SpeechConfig, the method defines a list of languages that the recogniser will support. In this case, the supported languages include English (en-US), French (fr-FR), and Italian (it-IT). These language codes are stored in a List¡String¿, and an AutoDetectSourceLanguageConfig object is created using the list. The purpose of this configuration is to enable automatic detection of the spoken language, ensuring that the speech recognition engine can adapt to the user's language without needing explicit input about the language being spoken. Once the language configuration is in place, the SpeechRecognizer object is instantiated with the SpeechConfig and AutoDetectSourceLanguageConfig passed as parameters. This object forms the core of the speech recognition functionality, responsible for interpreting and processing spoken input from the user.

To handle recognised speech, the recognizer.recognized.addEventListener() method is employed. This event listener listens for successful speech recognition results. If the recognised speech matches a known phrase or command, the result is processed further. When speech is successfully recognised (ResultReason.RecognizedSpeech), the method logs the recognised text for debugging purposes using Log.i(). The recognised text is then displayed to the user by updating a TextView widget (recognizedText). This is done by invoking runOnUiThread(), which ensures that UI updates occur on the main thread, as Android's UI components can only be modified from the main thread. Additionally, the recognised speech text is processed for specific commands by invoking two custom methods: processSpeechCommand() and updateImageViewBasedOnSpeech(). These methods further analyse the recognised text to execute specific actions, such as updating the robot's responses, gestures, or multimedia display based on the user's input. In cases where no speech is recognised (ResultReason.NoMatch), the recogniser logs that no match was found and updates the TextView to notify the user. This comprehensive process ensures that the application can understand and act upon spoken input in a dynamic and responsive manner, enhancing the user experience in interacting with the robot.

```
private void startContinuousRecognition() {  2 usages  new *
    try {
        if (recognizer == null) {
            initializeRecognizer();
        }
        recognizer.startContinuousRecognitionAsync();
        isListening = true;
        runOnUiThread(() -> {
            sttState.setText("Listening...");
            sttState.setVisibility(View.VISIBLE);
            buttonBack.setVisibility(View.VISIBLE);
            recognizedText.setVisibility(View.VISIBLE);
        });
    } catch (Exception e) {
        Log.e(TAG,  msg: "Error starting continuous recognition: " + e.getMessage());
    }
}
```

Figure 3.12: startContinuousRecognition

The startContinuousRecognition() method is designed to initiate continuous speech recognition for the robot, enabling it to listen for spoken commands or input in real-time. The method starts by ensuring that the speech recogniser (recogniser) is initialised before attempting to start the recognition process. If the recogniser is null, indicating that it hasn't been set up, the method calls initializeRecognizer() to configure the necessary speech recognition settings. Once the recogniser is confirmed to be initialised, the method invokes startContinuousRecognitionAsync(). This function begins the continuous speech recognition process asynchronously, allowing the robot to listen for speech without blocking the main UI thread. By running asynchronously, the speech recognition process can handle incoming speech in the background, providing a seamless user experience. This is critical in interactive applications where real-time user input must be processed without noticeable delays.

After starting the recognition process, the method sets a flag (isListening = true) to indicate that the robot is currently in a listening state. This flag can be used later to control the behavior of the robot or stop the recognition process when needed. For example, it helps manage cases where the robot might need to temporarily stop listening, such as when processing an action or responding to a user query. The method then employs runOnUiThread() to make updates to the user interface, ensuring that UI elements are modified on the main thread. Inside this block, the text for sttState is updated to reflect that the robot is now "Listening...", making the status clear to the user. The sttState text field is made visible, along with other UI elements such as the buttonBack and recognisedText widgets, which give visual feedback to the user that the robot is actively listening. In the event of any errors while starting the continuous recognition process, the method catches exceptions using a try-catch block. If an exception occurs, the method logs the error message for debugging purposes using Log.e(). This ensures that any issues with the recognition process are captured and can be addressed, improving the robustness of the application. Overall, this method is essential for enabling continuous speech recognition in the robot, allowing for uninterrupted interaction and real-time response to user input.

```java
private void sayText(String text, String voiceName) {  3 usages  new *
    handler.post(() -> {
        try {
            SpeechConfig config = SpeechConfig.fromSubscription(SUBSCRIPTION_KEY, SERVICE_REGION);
            config.setSpeechSynthesisVoiceName(voiceName);
            SpeechSynthesizer synthesizer = new SpeechSynthesizer(config);
            new Thread(() -> {
                try {
                    Thread.sleep( millis: 500);
                    //BuddySDK.UI.setMood(FacialExpression.HAPPY);
                    BuddySDK.UI.setLabialExpression(LabialExpression.SPEAK_HAPPY);
                } catch (InterruptedException e) {
                    throw new RuntimeException(e);
                }
            }).start();
            SpeechSynthesisResult result = synthesizer.SpeakText(text);
            BuddySDK.UI.setLabialExpression(LabialExpression.NO_EXPRESSION);
            result.close();
            synthesizer.close();
        } catch (Exception e) {
            Log.i( tag: "info",  msg: "Error: " + e.getMessage());
            e.printStackTrace();
        } finally {
            isProcessing = false;
        }
```

Figure 3.13: sayText

The sayText() method is designed to enable Buddy, the robot, to vocalise specific text using a particular voice profile. This function integrates speech synthesis functionality into the robot, allowing it to respond to users or provide information in a human-like manner. The method operates asynchronously and ensures that Buddy's interaction feels natural and fluid. The method first utilises a handler.post() call to run its contents on the main thread, which is necessary when updating the user interface or interacting with hardware components. Inside this block, the method begins by setting up the configuration required for speech synthesis. This is done by creating an instance of SpeechConfig from the subscription key (SUBSCRIPTION_KEY) and the service region (SERVICE_REGION). The subscription key and region are associated with the Microsoft Cognitive Services API, which powers the speech synthesis feature. This API is responsible for converting the given text into audible speech. The method also takes a voiceName parameter, which specifies the voice profile to be used for the speech synthesis. Voice profiles allow the robot to speak in different accents, languages, or even with various tonalities. This flexibility is essential for tailoring Buddy's interactions to diverse environments, enhancing the user experience.

Once the speech configuration is set, a SpeechSynthesizer object is created using the config. This object handles the actual text-to-speech process. Before synthesising the speech, the method starts a new thread using new Thread(() -¿ ...) to simulate facial expressions. In this case, it delays by 500 milliseconds to synchronise the robot's mouth movement and expressions with its speech. The BuddySDK.UI.setLabialExpression(LabialExpression.SPEAK_HAPPY) command is used to make Buddy's face display a happy expression while speaking, improving the emotional connection with users. The core functionality of the method is the call to synthesizer.SpeakText(text). This line triggers the text-to-speech conversion, where the SpeechSynthesizer converts the input text into audio. While Buddy is vocalising the text, it displays happy facial expressions to simulate a natural speaking experience. Once the speech synthesis is complete, the labial expression is reset to LabialExpression.NO_EXPRESSION to return Buddy's face to a neutral state. After the speech is synthesised, the SpeechSynthesisResult object is closed to free up resources, and the SpeechSynthesizer is also closed. Properly closing these objects is crucial to avoid memory leaks or unnecessary use of computational resources. If any errors occur during the speech synthesis process, the method catches the exceptions using a try-catch block. In the case of an exception, an error message is logged using Log.i() for debugging purposes, ensuring that any issues can be tracked and resolved. The method finally ensures that the isProcessing flag is set to false, allowing the robot to handle further interactions after the speech synthesis is complete.

```
private void moveBuddy(float speed, float distance, Runnable onSuccess) {
    Log.i(TAG, "Sending moveBuddy command: speed=" + speed + ", distance=" + distance);
    runOnUiThread(() -> {
        recognizedText.setText("Moving...");
        recognizedText.setVisibility(View.VISIBLE);
    });
    BuddySDK.USB.moveBuddy(speed, distance, new IUsbCommadRsp.Stub() {
        @Override
        public void onSuccess(String s) throws RemoteException {
            Log.i(TAG, "moveBuddy success: " + s);
            runOnUiThread(() -> {
                recognizedText.setText("Move successful");
                if (onSuccess != null) {
                    onSuccess.run();
                }
            });
        }
        @Override
        public void onFailed(String s) throws RemoteException {
            Log.i(TAG, "moveBuddy failed: " + s);
            runOnUiThread(() -> recognizedText.setText("Fail to advance"));
        }
    });
}
```

Figure 3.14: moveBuddy

The moveBuddy() method is responsible for controlling the physical movement of the Buddy robot. This method sends a command to move the robot at a specified speed and for a specified distance. The method incorporates both the user interface updates and interaction with Buddy's movement functionalities through the BuddySDK. The method begins by logging the parameters of the movement command using the Log.i() function. This is useful for debugging, as it provides real-time feedback in the log about the speed and distance that the robot is being instructed to move. Next, the method ensures that the user is aware of the robot's action by updating the user interface. The runOnUiThread() method is used to ensure that the UI changes are made on the main thread. Inside this block, the recognizedText view, which displays status updates to the user, is set to show the text "Moving..." and is made visible. This informs the user that Buddy is actively responding to the move command. The core of the method is the BuddySDK.USB.moveBuddy() function, which is part of the BuddySDK, and it directly communicates with the robot's hardware to perform the movement. The method sends two key parameters: speed (which controls how fast the robot should move) and distance (which defines how far Buddy should travel). The movement command is wrapped in an anonymous class that extends IUsbCommadRsp.Stub, which handles the success or failure of the movement operation. When the movement command is successful, the onSuccess() method is triggered, indicating that Buddy has completed the movement as instructed. The runOnUiThread() block is then used to update the user interface again, this time displaying the message "Move successful" on the recognizedText view to inform the user of the successful execution. If an additional action is required upon successful movement (such as chaining operations), the onSuccess.run() method is invoked, which can trigger further steps defined by the user in the form of a Runnable passed to the moveBuddy() method. Conversely, if the movement command fails for any reason, the onFailed() method is triggered. This method logs an error message using Log.i(), allowing developers or users to understand that the movement attempt was unsuccessful. The recognizedText view is updated to show the message "Fail to advance," signaling to the user that something went wrong with the robot's movement.

The method is designed to ensure smooth integration between the robot's movement and the user experience. By using runOnUiThread() to handle all UI updates, it maintains responsiveness, ensuring that the UI reflects the robot's state in real-time. The use of success and failure callbacks in the BuddySDK enables robust handling of movement commands, ensuring that Buddy's actions are reliably communicated to the user and any subsequent actions are appropriately managed.

```java
private void rotateBuddy(float rotspeed, float angle, Runnable onSuccess) {
    Log.i(TAG, "Sending rotateBuddy command: rotspeed=" + rotspeed + ", angle=" + angle);
    runOnUiThread(() -> {
        recognizedText.setText("Rotating...");
        recognizedText.setVisibility(View.VISIBLE);
    });
    BuddySDK.USB.rotateBuddy(rotspeed, angle, new IUsbCommadRsp.Stub() {
        @Override
        public void onSuccess(String s) throws RemoteException {
            Log.i(TAG, "rotateBuddy success: " + s);
            runOnUiThread(() -> {
                recognizedText.setText("Rotate successful");
                if (onSuccess != null) {
                    onSuccess.run();
                }
            });
        }
        @Override
        public void onFailed(String s) throws RemoteException {
            Log.i(TAG, "rotateBuddy failed: " + s);
            runOnUiThread(() -> recognizedText.setText("Fail to rotate"));
        }
    });
}
```

Figure 3.15: rotateBuddyt

The rotateBuddy() method is responsible for managing the rotation of the Buddy robot, allowing it to rotate by a specified angle at a defined rotational speed. This method integrates both user feedback through the interface and interactions with the robot's hardware via the BuddySDK. The method starts by logging the parameters of the rotation command, specifically the rotspeed (rotation speed) and angle (rotation angle), using Log.i(). This logging is important for debugging, as it provides real-time feedback regarding the specifics of the command being sent to the robot. To ensure that the user is informed about the current action, the method updates the user interface using runOnUiThread(). This is essential because all UI updates in Android must occur on the main thread. Inside this block, the recognizedText view, which is used to display status messages to the user, is updated to show the text "Rotating..." and is made visible. This provides immediate feedback to the user that Buddy is in the process of executing a rotation command.

The core of the method is the BuddySDK.USB.rotateBuddy() function, part of the BuddySDK, which handles the actual rotation of the robot. This method sends the two parameters—rotspeed and angle—to the robot. The rotational speed controls how fast Buddy rotates, and the angle specifies the degree to which the robot should turn. The method is wrapped inside an anonymous class that extends IUsbCommadRsp.Stub, which manages the callbacks for the success or failure of the rotation command. When the rotation is successful, the onSuccess() method is called, signaling that Buddy has completed the rotation as intended.

The runOnUiThread() block is used again to update the user interface, displaying the message "Rotate successful" on the recognizedText view. This informs the user that Buddy's rotation was executed correctly. Additionally, if any subsequent actions are needed after the rotation is completed (e.g., further tasks or chained operations), the onSuccess.run() method is invoked, allowing the user to define what should happen next through a Runnable passed to the rotateBuddy() method. In contrast, if the rotation command fails for any reason, the onFailed() method is triggered. This method logs an error message using Log.i(), indicating that the rotation attempt was unsuccessful. The recognizedText view is updated to display the message "Fail to rotate," notifying the user that an issue occurred during the robot's rotation.

The method ensures a seamless user experience by providing real-time feedback through the interface. The use of runOnUiThread() guarantees that the UI reflects the robot's actions in real time, maintaining responsiveness throughout the rotation process. The success and failure callbacks from the BuddySDK add reliability to the operation, allowing the system to handle various scenarios (successful execution or failure) effectively, with corresponding UI updates and possible follow-up actions.

## 3.8 Conclusion

The methodology involved developing three Android-based applications for the Buddy robot, targeting different aspects of multicultural education. Using the Buddy SDK and Microsoft Cognitive Services, the robot was programmed to interact through speech recognition, multilingual responses, and visual aids. The applications facilitated Buddy's participation in classroom activities, such as greeting children in multiple languages, engaging in symbolic play, and promoting cultural exchange. The design was modular, allowing for customization based on classroom diversity. The robot's movement, emotional expressions, and multimedia support were implemented to create an interactive learning environment tailored to children's linguistic and cultural needs.

# Chapter 4

# Experiment

## 4.1 Application Design and Development

The development of the Android-based applications for Buddy involved several stages, beginning with the conceptualisation of the robot's role within educational settings. The design of these applications was driven by the goal of creating an intuitive and interactive platform that could cater to the diverse needs of students in primary and early childhood education. As such, the applications are designed to be flexible, scalable, and capable of addressing various educational tasks, from language acquisition to intercultural pedagogy.

The user interface (UI) of the applications was developed with young learners in mind, featuring large, colourful buttons, simple navigation, and visual aids to guide interactions. The applications are designed to accommodate the varied cognitive abilities of young children, ensuring that all interactions with Buddy are intuitive and engaging. The apps allow teachers to customise Buddy's interactions based on the classroom's cultural or linguistic composition. For example, teachers can input student-specific data, such as their preferred language, so that Buddy can switch between languages when interacting with different students. This adaptability is essential for the growth of an inclusive learning environment, especially in multicultural classrooms where students may come from different linguistic and cultural backgrounds (100). To enable Buddy to perform the desired tasks, the applications are built upon the Buddy Software Development Kit (SDK), which provides a set of tools and libraries for controlling the robot's functions. The Buddy SDK is a critical component of the application's backend, enabling developers to access the robot's various functionalities, such as movement, speech, and multimedia display (101). For instance, when Buddy teaches a new word in a language-learning module, it can simultaneously display an image of the object on its screen, reinforcing the learning process through visual representation. In addition to static images, the multimedia display functionality allows for integrating dynamic content, such as animated videos or interactive games. This feature is particularly beneficial for engaging young learners, providing a multisensory learning experience. Students can interact with the content on Buddy's screen by selecting options, answering questions, or participating in interactive quizzes.

## 4.2 Implementation of App1

The App begins with the initialization of essential variables, such as the subscription key and service region, which are necessary for the speech recognition capabilities. The application also defines a path for video storage and sets up flags that help in monitoring the state of the app, such as whether the app is currently listening to user commands or processing speech input. Upon launching the application, the main function onCreate is called, which is responsible for setting up the user interface components, such as buttons, text views, and image views. These components are initialized through the initViews function, and the app listens for specific user interactions, like pressing buttons, through event listeners. For instance, when the user presses the "Listen" button, it triggers the speech recognition feature, which hides the welcome image and starts listening for voice commands. Additionally, the app verifies that all required permissions, particularly for accessing storage and using the device's microphone, are granted. If the permissions are not granted, the app requests them from the user.

The core functionality of the app lies in its ability to process speech commands and perform actions accordingly. Speech recognition is initialized using the initializeRecognizer function, which configures the speech recognizer to detect commands in multiple languages, including English, French, and Italian. When speech is recognized, the app processes the text in lowercase to ensure consistency in command recognition. This ensures that different variations of user commands, such as "move forward" or "turn left," are properly handled. Based on the recognized command, the app triggers the corresponding actions, such as moving Buddy forward or rotating the robot left or right. These commands are sent to Buddy's movement system, and the app provides visual feedback by updating the text views to inform the user of the success or failure of the movement commands. The application also supports multimedia interaction, specifically by allowing users to request video playback. If the user issues a command such as "show videos," the app loads video files from the specified directory and displays them as a list of thumbnails. The user can then select a video to play, and the app uses the playVideo function to launch an intent that plays the video file. The implementation is designed to handle cases where the file does not exist by providing error feedback to the user. Similarly, the app can remove the displayed video thumbnails when the user issues a command such as "remove videos," ensuring a clean and responsive user interface.

A critical aspect of the app's functionality is its ability to handle greetings in multiple languages. The app detects greetings like "hello," "ciao," or "bonjour," and responds in the corresponding language, showcasing Buddy's multilingual capabilities. This feature aligns with the app's goal of creating an interactive, multicultural environment where Buddy can engage with users in a personalized manner. For example, when a greeting in Italian is recognized, Buddy responds in Italian with a friendly message, enhancing the user's sense of connection with the robot. In addition to these features, the app ensures that commands are processed efficiently by using the isProcessing flag to prevent duplicate command handling. This ensures that the robot does not process multiple commands simultaneously, maintaining smooth and predictable interactions. When the app is not listening for speech commands, the stopListening function is invoked, halting the speech recognition process and updating the user interface to reflect the current state.

The flow of the application from initialization to execution follows a logical structure, where user commands are captured, processed, and acted upon in real-time. Each component of the app plays a role in ensuring that Buddy responds accurately to user input, whether it be moving, playing videos, or engaging in conversation. The app's design leverages Android functionalities such as permission handling, speech recognition, and multimedia playback to create an engaging user experience tailored to the specific scenario. The pseudocode, when translated into actual implementation, ensures that Robot Buddy operates seamlessly within the educational and interactive setting envisioned in the scenario.

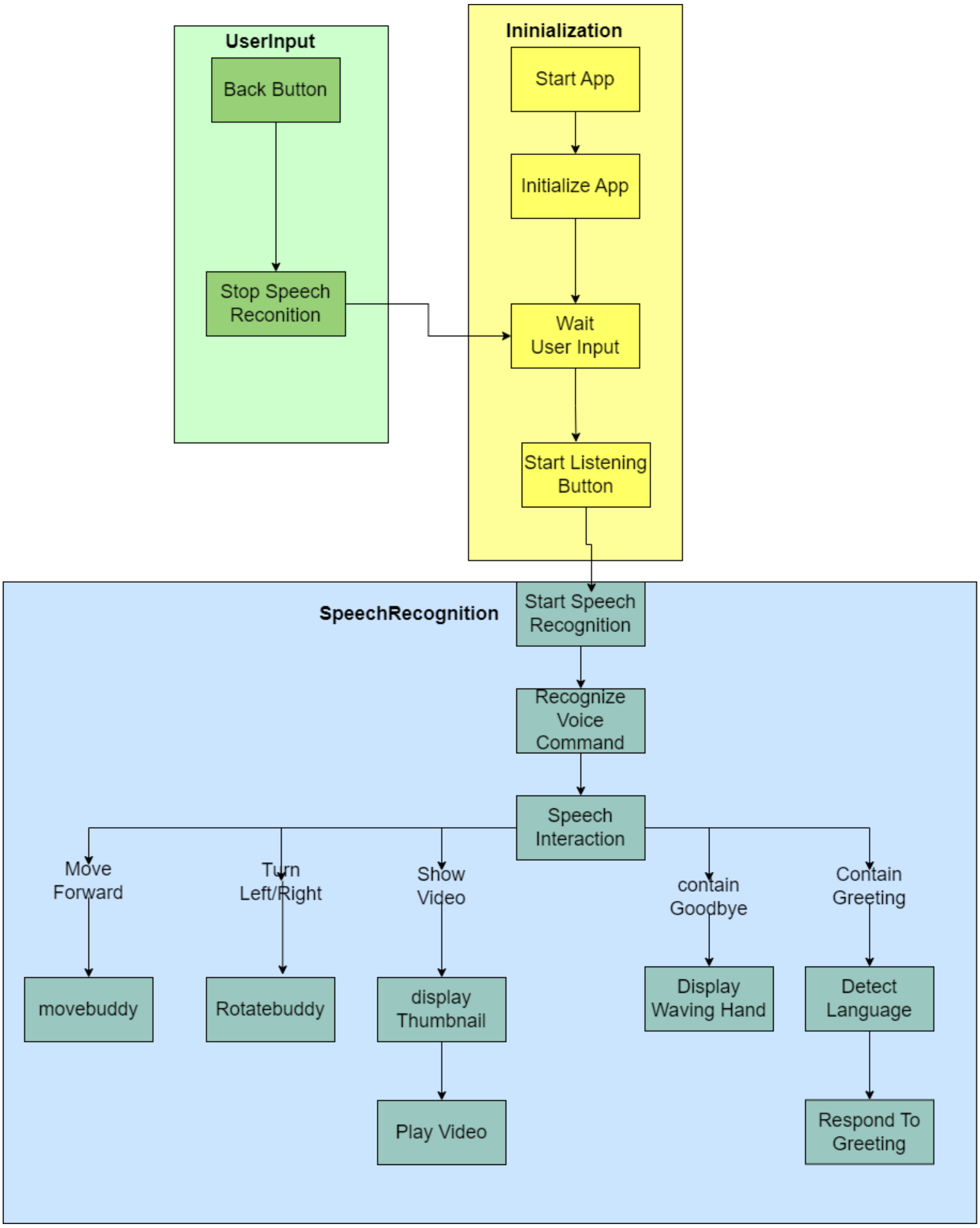


Figure 4.1: Flowchart of App1

## 4.3 Implementation of App2

The Android application interacts with the Buddy robot, focusing on enhancing user engagement through speech recognition, multimedia interactions, and robot movements. The main functionalities are centered on speech-driven commands and robot responses, where Buddy can move, rotate, and respond to user queries. The application starts by calling the onCreate() method, which initializes the user interface and sets up the necessary views for interaction. This is done through the initViews() function, where buttons, images, text fields, and other interface elements are connected to their respective components. The primary views include buttons for controlling speech recognition, a list view for displaying video files, and text fields for displaying recognized speech or status updates. Once these elements are in place, the application configures the listeners through configureListeners(), allowing for user interactions such as selecting videos or triggering speech recognition.

Speech recognition is a core component of this application, facilitated by the initializeRecognizer() function. Using Microsoft Cognitive Services, the application configures a speech recognition engine capable of detecting multiple languages. The speech recognition engine is set to continuously listen for user input when the startContinuousRecognition() function is invoked. This function also updates the user interface to indicate that the app is actively listening for commands, providing a visual cue for the user. When speech is recognized, the handleSpeechInteraction() function processes the spoken input. It first checks whether the app is already processing another input to prevent duplicate handling. The function then interprets specific commands related to movement, such as "move," "left," or "right," and executes the corresponding action by calling either moveBuddy() for forward movement or rotateBuddy() for rotation. These functions send commands to Buddy, making it move or rotate based on the recognized speech. Feedback is provided to the user in the form of status updates, such as "Moving" or "Rotating," which are displayed in the user interface.

In addition to movement commands, the app can handle cultural interactions and other conversational inputs based on the detected language. The app supports multiple languages, including English, French, and Italian, and switches its responses accordingly. For instance, if the user asks about Buddy's culture or makes references to cultural foods, the application responds with information about Buddy's home planet and favorite foods using the handleSpeechInteractionEN(), handleSpeechInteractionFR(), or handleSpeechInteractionIT() functions, depending on the current language. The robot can ask users to participate in games.

The speech-based interaction is further enhanced by the robot's emotional expressions. Buddy can display facial expressions such as "happy" or "listening" depending on the context of the conversation, adding a layer of emotional intelligence to the interaction. The setMoodTemporarily() function is responsible for setting these expressions, which remain for a few seconds before resetting to a neutral state. The sayText() function synthesizes speech to provide verbal feedback. Using text-to-speech capabilities, the app converts text into speech based on the recognized language, allowing Buddy to engage in more conversational exchanges with the user. For example, when the user mentions food or drink, Buddy responds with comments such as "Thank you! I would love some tea." This conversational ability makes the interaction feel more natural and human-like.

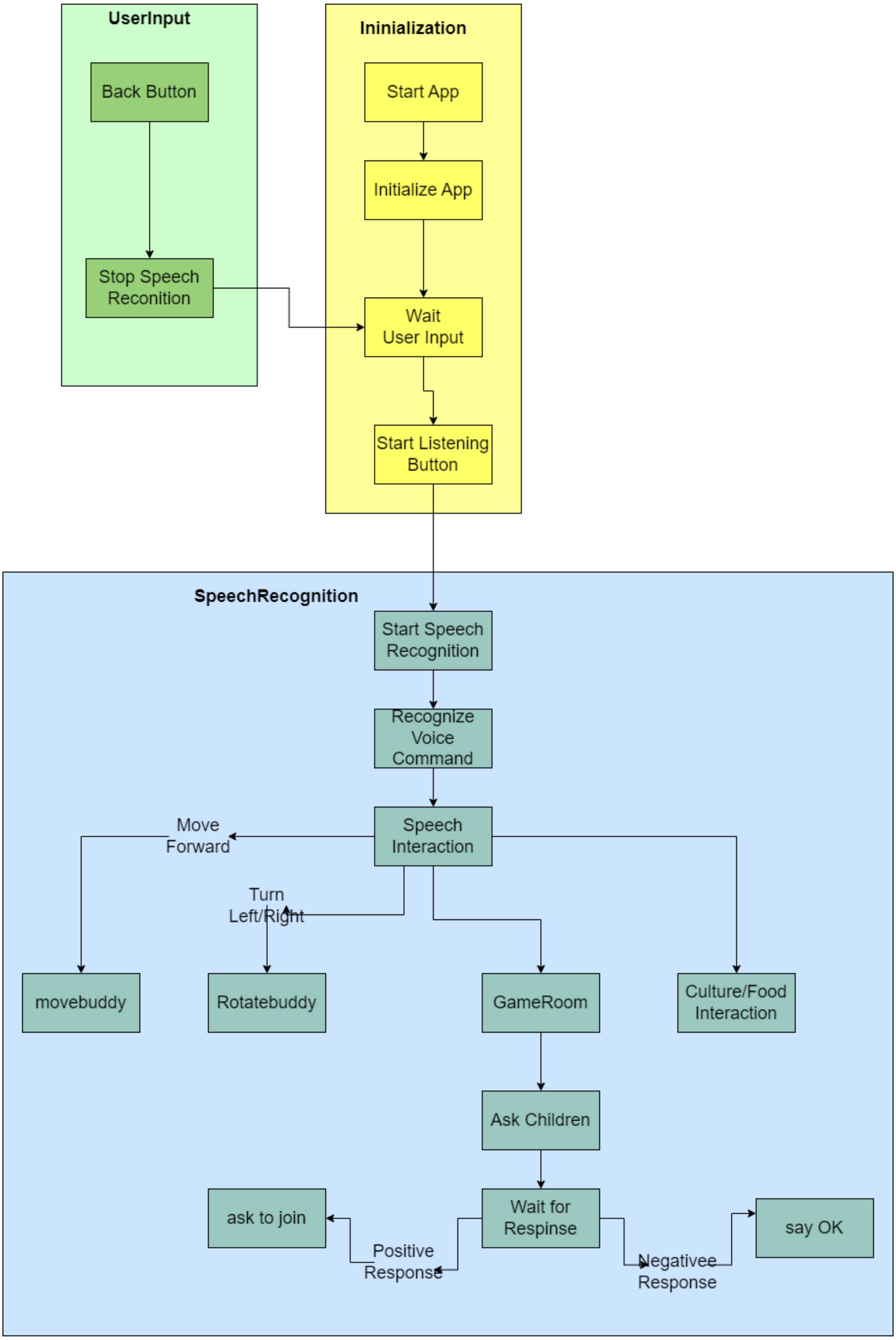


Figure 4.2: Flowchart App2

## 4.4 Implementation of App3

The implementation of an Android application designed for interaction with Buddy, a social robot that engages in activities such as speech recognition, movement, multimedia display, and interactive dialogues. This application harnesses several key functionalities to create an engaging environment where Buddy can interact with users through voice commands, multimedia presentations, and playful activities. When the application is launched, the OnCreate function is called to initialize the app and its views through the initViews() method. This involves setting up key elements like buttons, text fields, and image views to create a user interface for interaction. Once the views are initialized, listeners are configured to manage user actions, such as pressing buttons or selecting items from a list. This is done through the configureListeners() method, which associates specific behaviors with user actions. For example, pressing the "Listen" button hides certain elements of the interface and starts continuous speech recognition.

The next phase of the implementation involves checking and requesting permissions using the checkPermissions() function. Since the app requires access to storage and audio functionalities, it first verifies whether the necessary permissions have been granted by the user. If not, the app requests them; otherwise, it proceeds to load multimedia files stored on the device.

In addition to multimedia interaction, Buddy is also capable of movement, controlled by voice commands. The moveBuddy() function handles Buddy's movement by sending speed and distance commands to the robot, while the rotateBuddy() function controls its rotation. Both functions provide feedback to the user by updating the user interface to display messages like "Moving" or "Rotating," and they notify the user upon successful completion or failure of these actions. This combination of visual feedback and robot movement adds a dynamic element to the interaction, making it more tangible and responsive. Speech recognition forms a core component of the application, and this is implemented through the initializeRecognizer() and startContinuousRecognition() functions. These functions set up a speech recognition service using Microsoft's Cognitive Services and enable continuous listening for voice input. When speech is recognized, the app processes the input through the handleSpeechInteraction() function, which interprets the spoken commands and determines the appropriate response. If a movement command is recognized, Buddy responds by moving or rotating accordingly. If the speech relates to a game or cultural interaction, the app switches to handling those inputs based on the detected language.

The handleSpeechInteraction() function plays a pivotal role in enabling the robot to respond appropriately to different inputs. For instance, if the user mentions "move," Buddy will execute a movement command. If cultural interactions are detected, the application responds based on the current language setting (English, French, or Italian). Depending on the recognized language, the app calls different interaction functions such as handleGameRoomInteractionEN() for English, handleGameRoomInteractionFR() for French, and handleGameRoomInteractionIT() for Italian. These language-specific functions manage the flow of dialogue, asking questions about culture, games, or clothing in the appropriate language and awaiting user responses. The speech-based interaction is further enriched by the robot's emotional expressions and mood settings, which are controlled by the setMoodTemporarily() function. This function allows the robot to display various facial expressions such as "happy" or "listening" during the interaction, adding an emotional dimension to the robot's responses. The app can also provide spoken feedback through the sayText() function, which uses speech synthesis to generate natural-sounding responses in different languages. The timing of these responses is measured to ensure that the interactions feel natural and conversational.

An additional feature of the app is its game functionality, implemented through the showTabletGame() and simulateUserGuess() functions. The robot presents images to the user, asking them to guess the names of the items shown. After a delay, the app provides feedback on the user's guesses, either confirming correctness or offering the correct answer. This game-like interaction fosters learning and engagement, allowing users to interact with the robot in a playful and educational way.

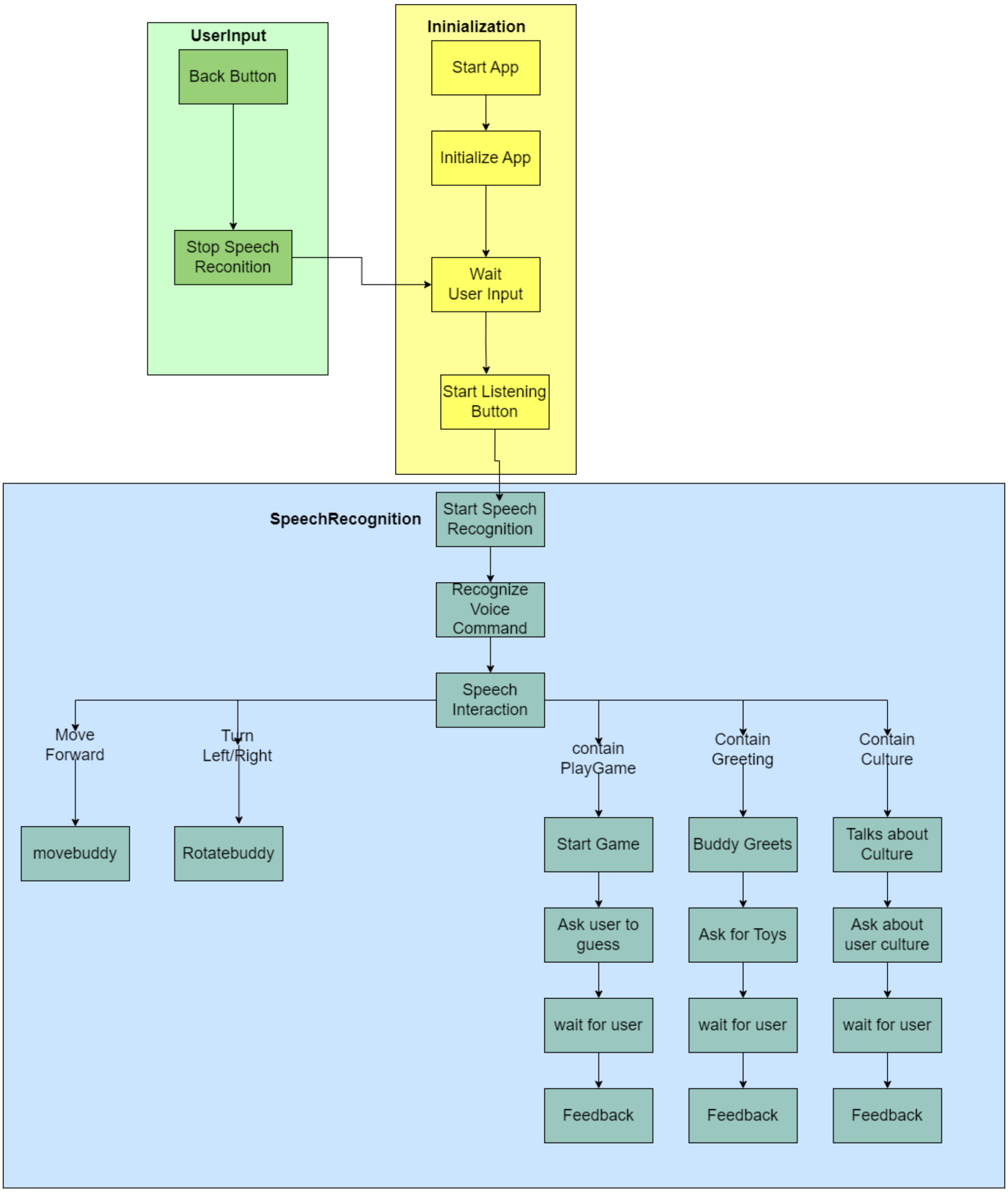


Figure 4.3: Flowchart App3

## 4.5 Evaluation of Scripts

Robot Buddy is designed to facilitate interactive, speech-based engagements, particularly in settings like multicultural environments such as kindergartens. These interactions are crafted to encourage curiosity, promote cultural sharing, and foster inclusion among children from diverse backgrounds. One of the most distinct aspects of Buddy's interactions involves its ability to initiate conversations about its own "culture," thereby encouraging children to share their cultural experiences in return. For example, when prompted by a child to describe its background, Buddy might say: "I am Buddy, a robot from another planet. On my planet, we wear shiny silver clothes, and we love playing games where we fly in the sky. What about your culture?" This interaction is intentionally crafted to spark children's curiosity and encourage them to reflect on their own cultural identity, leading to a broader exchange of traditions, games, and experiences. This type of interaction, especially within the context of a multicultural kindergarten, can have far-reaching effects. It creates a space where children from diverse cultural backgrounds can feel safe and encouraged to share aspects of their lives, whether it is about the clothing they wear or the games they play at home. For instance, if a child responds by describing their traditional clothing or favorite game, Buddy extends the conversation with a follow-up question such as, "Can you tell me what kind of clothes you wear? What games do you play?" This type of open-ended dialogue fosters an enriching learning environment where children not only feel validated but are also introduced to different cultures in a non-intimidating, friendly manner. The robot's design makes it approachable, and its consistent positive reinforcement encourages further interaction from the children.

In a multicultural kindergarten, interactions like these can serve as vital tools for promoting inclusivity and mutual respect. Children, particularly those who might be quieter or less inclined to speak about their differences in a traditional classroom setting, may find the robot's neutral and engaging behavior more comforting. As Buddy draws out conversations about traditions, games, and personal experiences, it provides a platform for all children, regardless of their background, to feel seen and heard. Moreover, by enabling children to share their cultures, Buddy promotes cross-cultural awareness and helps build a classroom culture of understanding and respect for diversity. Beyond cultural sharing, Buddy also engages in more playful and familiar conversations that focus on children's day-to-day experiences. For instance, Buddy might greet the children with, "Hello! What are you playing with? Can you show me your toys?" Such interactions are specifically designed to evoke curiosity from the children and prompt them to share their toys or ongoing play activities with Buddy. When a child mentions a toy or responds positively, Buddy follows up with responses like, "Wow, that looks fun! Can you teach me what it's called? How do you say that in your language?" These types of exchanges not only foster communication but also encourage language learning and cultural exchange. This can be particularly beneficial in a diverse classroom where children may speak multiple languages at home. The robot's interaction allows children to bridge these language barriers in an organic and engaging way, enhancing their sense of belonging. Buddy's ability to converse in multiple languages—such as English, French, and Italian—adds to its effectiveness as a tool for inclusivity. For example, when addressing a French-speaking child, Buddy might say, "Je suis Buddy, un robot d'une autre planète... Et ta culture?" Similarly, with an Italian-speaking child, Buddy might introduce itself by saying, "Sono Buddy, un robot di un altro pianeta... E la tua cultura?" These multilingual capabilities enable Buddy to adapt to the linguistic needs of the children, ensuring that non-native English speakers or children from bilingual homes feel included in the conversation. This feature helps Buddy become a powerful tool in creating a language-inclusive environment, where children from different linguistic backgrounds can participate actively without feeling alienated due to language barriers.

The positive aspects of such interactions are evident. First and foremost, Buddy helps create a space where children can celebrate their cultural and linguistic diversity. By encouraging children to share details about their lives, Buddy can help foster a deeper sense of inclusion and belonging, even among those who might otherwise feel marginalized. Furthermore, by offering interactions in multiple languages, Buddy ensures that children who speak languages other than English are not left out of important classroom conversations. This fosters an early appreciation of multilingualism and cultural exchange among children, which can have long-term positive effects on their social development. However, while Buddy's design offers many advantages, it is important to consider the potential limitations and challenges inherent in relying on a robot for cultural and social interactions. One potential issue is that some children, particularly those who are shy or less familiar with technology, might find it difficult to engage naturally with Buddy. While the robot is designed to be friendly and approachable, the fact that it is a machine might make interactions feel less organic compared to human-to-human communication. For these children, Buddy's scripted interactions, although engaging, might not fully replace the warmth and spontaneity of an interaction with a human teacher or peer. Additionally, despite its multilingual capabilities, Buddy's language database may still be limited in terms of dialects or less commonly spoken languages. This could result in frustration for children who speak minority languages that are not supported by Buddy's programming. While the robot can switch between major languages like English, French, and Italian, the absence of support for less common languages might inadvertently alienate some children, especially in diverse settings where the student population may be highly multicultural and multilingual.

Another potential downside is the over-reliance on technology for cultural and social exchanges. While Buddy's role in fostering cultural discussions and language learning is valuable, it is essential that these interactions are complemented by human guidance. Over-dependence on a robot for initiating conversations about culture or play may reduce opportunities for peer-to-peer interactions. Children learn important social cues, empathy, and cooperation through direct engagement with their classmates. If Buddy becomes the primary facilitator of these discussions, it could potentially hinder the natural development of these interpersonal skills. Human supervision is therefore a crucial element in ensuring that Buddy's presence is beneficial without becoming a substitute for personal interactions. Teachers must continue to play an active role in moderating and guiding these conversations, providing emotional support, and encouraging peer interactions. Additionally, teachers can use Buddy's interactions as a starting point for deeper classroom discussions, encouraging students to build on the conversations started by the robot. This way, Buddy enhances the learning environment rather than replacing the more organic and nuanced aspects of human interaction.

Despite these limitations, the carefully designed scripts of Buddy's interactions provide a valuable framework for promoting cultural diversity, language learning, and social inclusion in a kindergarten setting. By acting as a bridge between children from different backgrounds, Buddy creates opportunities for cultural exchange and mutual understanding. For example, when Buddy asks children to describe their traditional clothing or favorite games, it invites children to reflect on their own identity while learning about their peers. Similarly, by encouraging children to teach Buddy words in their own language, the robot promotes language sharing and cross-cultural communication, which are vital skills in an increasingly globalized world. Moreover, the structured yet flexible nature of Buddy's interactions makes it a versatile tool in the classroom. The robot can initiate conversations, recognize responses, and guide interactions in ways that promote not only cultural awareness but also cognitive development. The playful tone of its interactions, combined with its ability to switch between languages, allows Buddy to engage children in meaningful learning experiences without making them feel like they are being tested or judged. This non-threatening and supportive environment can be particularly beneficial for young children as they navigate complex social and cultural dynamics.

# Chapter 5

# Conclusions

## 5.1 Overview

The conclusion chapter includes a detailed summary of the study's findings, focusing on how the Buddy robot successfully met the objectives of promoting multilingual engagement, cultural exploration, and cultural exchange in early childhood education. It presents the results of the experiments, highlighting Buddy's effectiveness in creating an inclusive learning environment. Additionally, the chapter addresses the limitations encountered during the experiments, such as language processing challenges, and discusses the overall impact of integrating social robots into multicultural classrooms. It concludes with the broader implications of the study, suggesting that social robotics can significantly enhance modern educational practices.

## 5.2 Conclusion

The experiments demonstrated that Buddy could engage with children in multiple languages through songs, visuals, and greetings. In the lab, Buddy recognized and responded to greetings in different languages, creating an atmosphere where users from various cultural and linguistic backgrounds felt acknowledged. This approach not only supported language development but also reduced language barriers, fostering a sense of belonging among the children. In the second scenario, Buddy's involvement in symbolic play and discussions on cultural traditions showed significant success in promoting cultural exploration. During the experiments, Buddy interacted with user by asking questions about their cultural practices and introducing elements from its own "imagined culture." By simulating gestures, using visual aids on its tablet, and engaging in playful dialogue, Buddy helped user explore and appreciate cultural diversity.

Further, the third objective focused on fostering cultural exchange through storytelling and interactive activities. The experiments showed that Buddy could successfully facilitate discussions on cultural topics, such as traditional clothing and games, prompting children to describe aspects of their cultures. By using its language recognition and translation features, Buddy ensured every child could participate, regardless of their linguistic background. Additionally, the robot's use of games, where user guessed the names of cultural objects, reinforced cultural awareness and made the exchange both interactive and educational. The robot's ability to switch between languages, simulate cultural gestures, and participate in cultural dialogue can significantly improve students' linguistic and intercultural learning experiences. These results support the thesis's assertion that integrating social robots into multicultural education can address linguistic and cultural barriers, offering a dynamic and equitable learning experience.

## 5.3 Limitations

One notable limitation was Buddy's limited proficiency in less common languages, which could present challenges in highly diverse classrooms. The robot's dependency on speech recognition accuracy occasionally led to misinterpretations, particularly in noisy environments or with accents, highlighting the need for more advanced and adaptable language-processing capabilities. Additionally, while Buddy effectively used its tablet to simulate gestures and expressions, the lack of physical limb movements limited the realism of some interactions.

## 5.4 Recommendations

One of the most critical aspects to enhance in future iterations of social robots like Buddy is its multilingual speech recognition capabilities. This feature is paramount for fostering an inclusive environment in classrooms with students from diverse linguistic backgrounds. Currently, Buddy's speech recognition system is effective but has limitations when processing less commonly spoken languages and dialects, often resulting in misinterpretations that hinder the robot's ability to fully engage with every student. The primary recommendation is to integrate an advanced, context-aware speech recognition system, specifically trained on a broader array of languages and dialects. This can be achieved by leveraging machine learning models that incorporate extensive datasets from various linguistic sources, including regional accents and variations in pronunciation. Collaboration with language experts and educators will be crucial in curating a diverse linguistic dataset that accurately reflects the classroom's multicultural environment. Additionally, adopting continuous learning algorithms would allow Buddy to adapt and improve its recognition accuracy over time, learning from its interactions with students. Implementing this enhancement would significantly improve Buddy's effectiveness in classrooms. By accurately recognizing and responding in a child's native language, Buddy can create a more welcoming atmosphere, encouraging students to actively participate and express themselves. This is particularly important for students who may experience anxiety or exclusion due to language barriers. Furthermore, accurate multilingual engagement supports language acquisition for both native and second-language learners, making Buddy a valuable tool in developing linguistic proficiency and intercultural communication skills.

# Bibliography


[1] C. Lin. Understanding cultural diversity and diverse identities. *Quality education*, pages 929–938, 2020. 1

[2] D. Malinowski, H. H. Maxim, and S. Dubreil. *Language Teaching in the Linguistic Landscape*. Springer International Publishing, 2020. 1

[3] M. T. Hora and S. B. Millar. *A guide to building education partnerships: Navigating diverse cultural contexts to turn challenge into promise*. Taylor Francis, 2023. 1

[4] B. F. Gutiérrez, M. R. Glimäng, S. Sauro, and R. O'Dowd. Preparing students for successful online intercultural communication and collaboration in virtual exchange. *Journal of International Students*, 12(S3):149–167, 2022. 1

[5] C. A. Eden, O. N. Chisom, and I. S. Adeniyi. Cultural competence in education: strategies for fostering inclusivity and diversity awareness. *International Journal of Applied Research in Social Sciences*, 6(3):383–392, 2024. 1

[6] M. Schwarzenthal, M. K. Schachner, L. P. Juang, and F. J. Van De Vijver. Reaping the benefits of cultural diversity: Classroom cultural diversity climate and students' intercultural competence. *European Journal of Social Psychology*, 50(2):323–346, 2020. 2

[7] O. M. Williams-Duncan. Cross-cultural community building: Strategies for improving interactions in diverse educational settings. *Multicultural Education*, 27(2):18–23, 2020. 2

[8] S. Amin. Diversity enforces social exclusion: Does exclusion never cease? *Journal of Social Inclusion*, 10(1), 2019. 2

[9] H. Parkhouse, C. Y. Lu, and V. R. Massaro. Multicultural education professional development: A review of the literature. *Review of Educational Research*, 89(3):416–458, 2019. 2

[10] T. Belpaeme and F. Tanaka. *Social robots as educators*, page 143. OECD Publishing, Paris, 2021. 2

[11] I. Papadopoulos, R. Lazzarino, S. Miah, T. Weaver, B. Thomas, and C. Koulouglioti. A systematic review of the literature regarding socially assistive robots in pre-tertiary education. *Computers Education*, 155:103924, 2020. 2, 7, 16

[12] R. Shadiev and J. Yu. Review of research on computer-assisted language learning with a focus on intercultural education. *Computer Assisted Language Learning*, 37(4):841–871, 2024. 2

[13] A. Gubenko, C. Kirsch, J. N. Smilek, T. Lubart, and C. Houssemand. Educational robotics and robot creativity: An interdisciplinary dialogue. *Frontiers in Robotics and AI*, 8:662030, 2021. 2, 11

[14] A. Alam. Social robots in education for long-term human-robot interaction: Socially supportive behaviour of robotic tutor for creating robo-tangible learning environment in a guided discovery learning interaction. *ECS Transactions*, 107(1):12389, 2022. 2

[15] A. G. Richards. *Cultural diversity, conceptual pedagogy, and educating students for their futures*, pages 183–205. Springer Singapore, 2019. 3, 4

[16] K. Amara. The role of foreign language teaching in enhancing students' intercultural competence. *Near East University Online Journal of Education*, 3(1):22–36, 2020. 3, 4

[17] K. C. Krebs. Global, international, and intercultural learning in university classrooms across the disciplines. *Research in Comparative and International Education*, 15(1):36–51, 2020. 3, 4

[18] H. E. Hoff. The evolution of intercultural communicative competence: Conceptualisations, critiques, and consequences for 21st-century classroom practice. *Intercultural Communication Education*, 3(2):55–74, 2020. 3, 4

[19] J. Landsman and C. W. Lewis. *White teachers/diverse classrooms: Creating inclusive schools, building on students' diversity, and providing true educational equity*. Taylor Francis, 2023. 3, 5

[20] S. Higgen and M. Mösko. Mental health and cultural and linguistic diversity as challenges in school? an interview study on the implications for students and teachers. *PloS One*, 15(7):e0236160, 2020. 3, 4

[21] K. Markey, B. O'Brien, C. Kouta, C. Okantey, and C. O'Donnell. Embracing classroom cultural diversity: Innovations for nurturing inclusive intercultural learning and culturally responsive teaching. *Teaching and Learning in Nursing*, 16(3):258–262, 2021. 3, 5

[22] J. Fong. An evaluation of an education abroad program on the intercultural learning and cross-cultural adaptability skills of university undergraduates. *Higher Education Evaluation and Development*, 14(2):55–68, 2020. 3, 4

[23] K. Markey, M. M. Graham, D. Tuohy, J. McCarthy, C. O'Donnell, T. Hennessy, and B. O'Brien. Navigating learning and teaching in expanding culturally diverse higher education settings. *Higher Education Pedagogies*, 8(1):2165527, 2023. 3, 4

[24] M. Tambyah. Intercultural understanding through a 'similar but different' international teaching practicum. *Teaching Education*, 30(1):105–122, 2019. 3, 5

[25] A. Othman and N. Ruslan. Intercultural communication experiences among students and teachers: implication to in-service teacher professional development. *Journal for Multicultural Education*, 14(3/4):223–238, 2020. 4

[26] A. Akkari and M. Radhouane. *Intercultural approaches to education: From theory to practice*. Springer Nature, 2022. 4, 5

[27] H. Woo, G. K. LeTendre, T. Pham-Shouse, and Y. Xiong. The use of social robots in classrooms: A review of field-based studies. *Educational Research Review*, 33:100388, 2021. 6, 7, 8, 12, 13

[28] M. Donnermann, P. Schaper, and B. Lugrin. Integrating a social robot in higher education–a field study. In *2020 29th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*, pages 573–579, 2020. 6

[29] M. H. Batubara and M. G. S. Mahardhika. An analysis on students' difficulties in changing active to passive voice. *Jurnal As-Salam*, 4(1):61–78, 2020. 6

[30] L. Darling-Hammond, L. Flook, C. Cook-Harvey, B. Barron, and D. Osher. Implications for educational practice of the science of learning and development. *Applied Developmental Science*, 24(2):97–140, 2020. 6

[31] K. Youssef, S. Said, S. Alkork, and T. Beyrouthy. A survey on recent advances in social robotics. *Robotics*, 11(4):75, 2022. 7

[32] S. Mohammed and L. Kınyo. Constructivist theory as a foundation for the utilization of digital technology in the lifelong learning process. *Turkish Online Journal of Distance Education*, 21(4):90–109, 2020. 7

[33] I. M. Verner, H. Perez, and R. Lavi. Characteristics of student engagement in high-school robotics courses. *International Journal of Technology and Design Education*, 32(4):2129–2150, 2022. 7

[34] A. Riedmann, P. Schaper, and B. Lugrin. Integration of a social robot and gamification in adult learning and effects on motivation, engagement, and performance. *AI Society*, 39(1):369–388, 2024. 7

[35] R. Van den Berghe, J. Verhagen, O. Oudgenoeg-Paz, S. Van der Ven, and P. Leseman. Social robots for language learning: A review. *Review of Educational Research*, 89(2):259–295, 2019. 7, 9

[36] J. E. Hernandez, N. Vasan, S. Huff, and C. Melovitz-Vasan. Learning styles/preferences among medical students: Kinesthetic learner's multimodal approach to learning anatomy. *Medical Science Educator*, 30(4):1633–1638, 2020. 7

[37] G. A. Papakostas, G. K. Sidiropoulos, C. I. Papadopoulou, E. Vrochidou, V. G. Kaburlasos, M. T. Papadopoulou, and N. Dalivigkas. Social robots in special education: A systematic review. *Electronics*, 10(12):1398, 2021. 7, 9

[38] M. Alghamdi, N. Alhakbani, and A. Al-Nafjan. Assessing the potential of robotics technology for enhancing education for children with autism spectrum disorder. *Behavioral Sciences*, 13(7):598, 2023. 7

[39] B. Louie, E. A. Björling, and A. C. Kuo. The desire for social robots to support english language learners: Exploring robot perceptions of teachers, parents, and students. In *Frontiers in Education*, volume 6, page 566909, 2021. 7, 8

[40] F. Wang and A. C. Cheung. Robots' social behaviors for language learning: A systematic review and meta-analysis. *Review of Educational Research*, 2024. 7, 8, 9

[41] J. C. Liang and G. J. Hwang. A robot-based digital storytelling approach to enhancing efl learners' multimodal storytelling ability and narrative engagement. *Computers Education*, 201:104827, 2023. 8

[42] J. Kormos and A. M. Smith. *Teaching languages to students with specific learning differences*, volume 18. Channel View Publications, 2023. 8

[43] S. Tellex, N. Gopalan, H. Kress-Gazit, and C. Matuszek. Robots that use language. *Annual Review of Control, Robotics, and Autonomous Systems*, 3(1):25–55, 2020. 8

[44] M. Marge, C. Espy-Wilson, N. G. Ward, A. Alwan, Y. Artzi, M. Bansal, and Z. Yu. Spoken language interaction with robots: Recommendations for future research. *Computer Speech Language*, 71:101255, 2022. 8

[45] N. I. Arshad, A. S. Hashim, M. M. Ariffin, N. M. Aszemi, H. M. Low, and A. A. Norman. Robots as assistive technology tools to enhance cognitive abilities and foster valuable learning experiences among young children with autism spectrum disorder. *IEEE Access*, 8:116279–116291, 2020. 9

[46] A. M. Alcorn, E. Ainger, V. Charisi, S. Mantinioti, S. Petrović, B. R. Schadenberg, and E. Pellicano. Educators' views on using humanoid robots with autistic learners in special education settings in england. *Frontiers in Robotics and AI*, 6:107, 2019. 9

[47] D. E. Logan, C. Breazeal, M. S. Goodwin, S. Jeong, B. O'Connell, D. Smith-Freedman, and P. Weinstock. Social robots for hospitalized children. *Pediatrics*, 144(1), 2019. 9

[48] S. Yanuarini. Designing learning activities for improving social and imitation skills for children with autism spectrum disorder using a humanoid robot. Master's thesis, University of Twente, 2024. 9

[49] Y. Zhang, W. Song, Z. Tan, H. Zhu, Y. Wang, C. M. Lam, and L. Yi. Could social robots facilitate children with autism spectrum disorders in learning distrust and deception? *Computers in Human Behavior*, 98:140–149, 2019. 9

[50] A. Puglisi, T. Caprì, L. Pignolo, S. Gismondo, P. Chilà, R. Minutoli, and G. Pioggia. Social humanoid robots for children with autism spectrum disorders: a review of modalities, indications, and pitfalls. *Children*, 9(7):953, 2022. 9

[51] J. DiPietro, A. Kelemen, Y. Liang, and C. Sik-Lanyi. Computer-and robot-assisted therapies to aid social and intellectual functioning of children with autism spectrum disorder. *Medicina*, 55(8):440, 2019. 10

[52] S. Ali, N. Devasia, H. W. Park, and C. Breazeal. Social robots as creativity eliciting agents. *Frontiers in Robotics and AI*, 8:673730, 2021. 10

[53] C. Y. Andy and R. S. Masters. Improving motor skill acquisition through analogy in children with autism spectrum disorders. *Psychology of Sport and Exercise*, 41:63–69, 2019. 10

[54] R. Martin and J. Wilkins. Creating visually appropriate classroom environments for students with autism spectrum disorder. *Intervention in School and Clinic*, 57(3):176–181, 2022. 10

[55] S. Rasouli, G. Gupta, E. Nilsen, and K. Dautenhahn. Potential applications of social robots in robot-assisted interventions for social anxiety. *International Journal of Social Robotics*, 14(5):1–32, 2022. 10

[56] H. Chen, H. W. Park, and C. Breazeal. Teaching and learning with children: Impact of reciprocal peer learning with a social robot on children's learning and emotive engagement. *Computers Education*, 150:103836, 2020. 10, 11

[57] W. Yang, H. Luo, and J. Su. Towards inclusiveness and sustainability of robot programming in early childhood: Child engagement, learning outcomes and teacher perception. *British Journal of Educational Technology*, 53(6):1486–1510, 2022. 10, 12

[58] A. Haleem, M. Javaid, M. A. Qadri, and R. Suman. Understanding the role of digital technologies in education: A review. *Sustainable Operations and Computers*, 3:275–285, 2022. 11, 12

[59] V. G. Al Hakim, S. H. Yang, M. Liyanawatta, J. H. Wang, and G. D. Chen. Robots in situated learning classrooms with immediate feedback mechanisms to improve students' learning performance. *Computers Education*, 182:104483, 2022. 11

[60] S. A. Denham and K. H. Liverette. *The emotional basis of learning and development in early childhood education*, pages 43–64. Routledge, 2019. 11

[61] G. Peretti, F. Manzi, C. Di Dio, A. Cangelosi, P. L. Harris, D. Massaro, and A. Marchetti. Can a robot lie? young children's understanding of intentionality beneath false statements. *Infant and Child Development*, 32(2):e2398, 2023. 11

[62] B. Zohuri and F. Mossavar-Rahmani. Revolutionizing education: The dynamic synergy of personalized learning and artificial intelligence. *International Journal of Advanced Engineering and Management Research*, 9(1):143–153, 2024. 11

[63] I. Jawaid, M. Y. Javed, M. H. Jaffery, A. Akram, U. Safder, and S. Hassan. Robotic system education for young children by collaborative-project-based learning. *Computer Applications in Engineering Education*, 28(1):178–192, 2020. 11

[64] P. K. Keith, F. R. Sullivan, and D. Pham. *Roles, collaboration, and the development of computational thinking in a robotics learning environment*, pages 223–245. 2019. 11

[65] L. Cerna, C. Mezzanotte, A. Rutigliano, O. Brussino, P. Santiago, F. Borgonovi, and C. Guthrie. *Promoting inclusive education for diverse societies: A conceptual framework*. 2021. 12, 13

[66] V. Lim, M. Rooksby, and E. S. Cross. Social robots on a global stage: Establishing a role for culture during human–robot interaction. *International Journal of Social Robotics*, 13(6):1307–1333, 2021. 12

[67] C. S. Abacioglu, M. Volman, and A. H. Fischer. Teachers' multicultural attitudes and perspective-taking abilities as factors in culturally responsive teaching. *British Journal of Educational Psychology*, 90(3):736–752, 2020. 12, 13

[68] J. A. Banks and C. A. M. Banks. *Multicultural Education: Issues and Perspectives*. John Wiley Sons, 2019. 12, 13

[69] B. Louie, E. A. Björling, A. C. Kuo, and P. Alves-Oliveira. Designing for culturally responsive social robots: An application of a participatory framework. *Frontiers in Robotics and AI*, 9:983408, 2022. 12, 16

[70] T. Gnanasekar. User experience design and evaluation of persuasive social robot as language tutor at university: Design and learning experiences from design research. Master's thesis, Master's thesis, 2023. 13

[71] S. Probine and J. Perry. Does the 'more knowledgeable other' and the established discourses that accompany it have a place in ec today?: Rethinking and re-casting vygotsky for the twenty-first century. *Early Education*, 67:5–13, 2021. 13

[72] J. Joy, A. Bartolini, M. Milella, A. Sgorbissa, and C. T. Recchiuto. Social robotics and intercultural pedagogy. In *2024 32nd Mediterranean Conference on Control and Automation (MED)*, pages 292–297, 2024. 13

[73] C. Papadopoulos, N. Castro, A. Nigath, R. Davidson, N. Faulkes, R. Menicatti, and A. Sgorbissa. The caresses randomised controlled trial: exploring the health-related impact of culturally competent artificial intelligence embedded into socially assistive robots and tested in older adult care homes. *International Journal of Social Robotics*, 14(1):245–256, 2022. 14

[74] B. Bruno, N. Y. Chong, H. Kamide, S. Kanoria, J. Lee, Y. Lim, and A. Sgorbissa. *The CARESSES EU-Japan project: Making assistive robots culturally competent*, pages 151–169. Springer International Publishing, 2019. 14

[75] M. Obaid, R. Aylett, W. Barendregt, C. Basedow, Lee J. Corrigan, Lynne Hall, Aidan Jones, Arvid Kappas, Dennis Kuster, Ana Paiva, Fotis Papadopoulos, Sofia Serholt, and Ginevra Castellano. Endowing a robotic tutor with empathic qualities: Design and pilot evaluation. *International Journal of Humanoid Robotics*, 15(6), 2018. 14

[76] Y. Wang, L. Hespanhol, and M. Tomitsch. How can autonomous vehicles convey emotions to pedestrians? a review of emotionally expressive non-humanoid robots. *Multimodal Technologies and Interaction*, 5(12):84, 2021. 14

[77] I. Trifirò, A. Augello, U. Maniscalco, G. Pilato, F. Vella, and R. Meo. How are you? how a robot can learn to express its own roboceptions. *Procedia Computer Science*, 176:480–489, 2020. 14

[78] C. J. Sutherland and B. MacDonald. Robolang: a simple domain-specific language to script robot interactions. In *2019 16th International Conference on Ubiquitous Robots (UR)*, pages 265–270. IEEE, 2019. 14

[79] U. S. K. Rajapaksha, C. Jayawardena, and B. A. MacDonald. Ros-based heterogeneous multiple robots control using high-level user instructions. In *TENCON 2021-2021 IEEE Region 10 Conference (TENCON)*, pages 163–168. IEEE, 2021. 14

[80] T. Sawaragi, A. Oyama, S. Chida, M. Hayashi, and N. Sumi. Mitsubishi electric, 2024. 15

[81] Y. Umeda and S. Namematsu. Mitsubishi electric, 2020. 15

[82] W. M. Wang and H. J. Chang. Mitsubishi electric's robot teaching course design. In *2020 International Conference on Mathematics and Computers in Science and Engineering (MACISE)*, pages 223–228. IEEE, 2020. 15

[83] T. Shibata and K. L. Shum. Platform approach to open cyber physical system: Factory automation in japan. *Journal of Strategic Management Studies*, 13(1):35–46, 2021. DOI not available. 15

[84] P. S. Mohammed and E. 'Nell' Watson. *Towards inclusive education in the age of artificial intelligence: Perspectives, challenges, and opportunities*, pages 17–37. 2019. 15

[85] G. Head. Ethics in educational research: Review boards, ethical issues, and researcher development. *European Educational Research Journal*, 19(1):72–83, 2020. 15

[86] I. Villa. Humans and non-humans: Representation of diversity and exclusionary practices in twenty-first century british science fiction tv series, 2021. 15

[87] T. L. Andrews. Project sci-fi: Inviting aliens and robots into the english classroom, how fictional cultures can promote development of intercultural competence. Master's thesis, Master's thesis, 2020. 15

[88] M. L. Ornelas, G. B. Smith, and M. Mansouri. Redefining culture in cultural robotics. *AI SOCIETY*, 38(2):777–788, 2023. 16

[89] L. E. Johannessen, E. B. Rasmussen, and M. Haldar. Student at a distance: Exploring the potential and prerequisites of using telepresence robots in schools. *Oxford Review of Education*, 49(2):153–170, 2023. 16

[90] J. P. Boada, B. R. Maestre, and C. T. Genís. The ethical issues of social assistive robotics: A critical literature review. *Technology in Society*, 67:101726, 2021. 16

[91] R. Belk. Ethical issues in service robotics and artificial intelligence. *The Service Industries Journal*, 41(13-14):860–876, 2021. 16

[92] B. De Carolis, G. Palestra, C. Della Penna, M. Cianciotta, and A. Cervelione. Social robots supporting the inclusion of unaccompanied migrant children: Teaching the meaning of culture-related gestures. *Journal of e-Learning and Knowledge Society*, 15(2), 2019. 16

[93] P. J. Antony and F. Vaughn-Shavuo. *Creating culturally competent teachers in higher education*. Springer Briefs in Education. 2022. 16

[94] Muhammad Ayoub Kamal, Hafiz Wahab Raza, Muhammad Mansoor Alam, and M Mohd. Highlight the features of aws, gcp and microsoft azure that have an impact when choosing a cloud service provider. *Int. J. Recent Technol. Eng*, 8(5):4124–4232, 2020. 20

[95] Harsh Chawla and Hemant Kathuria. *Building microservices applications on Microsoft azure: designing, Developing, Deploying, and Monitoring*. Apress, 2019. 20

[96] Laura Romeo, Roberto Marani, Matteo Malosio, Anna G Perri, and Tiziana D'Orazio. Performance analysis of body tracking with the microsoft azure kinect. In *2021 29th Mediterranean Conference on Control and Automation (MED)*, pages 572–577. IEEE, 2021. 20

[97] Bulbul Gupta, Pooja Mittal, and Tabish Mufti. A review on amazon web service (aws), microsoft azure & google cloud platform (gcp) services. In *Proceedings of the 2nd International Conference on ICT for Digital, Smart, and Sustainable Development, ICIDSSD 2020, 27-28 February 2020, Jamia Hamdard, New Delhi, India*, 2021. 20

[98] Prakarsh Kaushik, Ashwin Murali Rao, Devang Pratap Singh, Swati Vashisht, and Shubhi Gupta. Cloud computing and comparison based on service and performance between amazon aws, microsoft azure, and google cloud. In *2021 International Conference on Technological Advancements and Innovations (ICTAI)*, pages 268–273. IEEE, 2021. 20

[99] Derya Ucuz et al. Comparison of the iot platform vendors, microsoft azure, amazon web services, and google cloud, from users' perspectives. In *2020 8th international symposium on digital forensics and security (ISDFS)*, pages 1–4. IEEE, 2020. 20

[100] I. Noel, Y. S. Rezeki, E. F. Rahmani, U. Salam, and W. Wardah. Developing android-based application "speak buddy" to improve senior high school student's speaking skill. *Jurnal Cakrawala Ilmiah*, 2(5):1957–1962, 2023. 42

[101] C. P. Adwinda and S. Pradono. Developing an android-based running application. *Journal of Critical Reviews*, 7(8):851–857, 2020. 42