



Usability Enhancement and Functional Extension of a Digital Tool for Rapid Assessment of Risk for Autism Spectrum Disorders in Toddlers Based on Pilot Test and Interview Data

Deeksha Adiani¹, Michael Schmidt¹, Joshua Wade¹(✉),
Amy R. Swanson², Amy Weitlauf², Zachary Warren³,
and Nilanjan Sarkar¹

¹ Adaptive Technology Consulting, LLC, Murfreesboro, TN 37127, USA
josh@innovateatc.com

² Treatment and Research Institute for Autism Spectrum Disorders (TRIAD),
Vanderbilt University, Nashville, TN 37212, USA

³ Pediatric, Psychiatry and Special Education, Vanderbilt University, Nashville,
TN 37212, USA

Abstract. Early accurate identification and treatment of young children with Autism Spectrum Disorder (ASD) represents a pressing public health and clinical care challenge. Unfortunately, large numbers of children are still not screened for ASD, waits for specialized diagnostic assessment can be very long, and the average age of diagnosis in the US remains between 4 to 5 years of age. In a step towards meaningfully addressing this issue, we previously developed Autoscreen: a digital tool for accurate and time-efficient screening, diagnostic triage, referral, and treatment engagement of young children with ASD concerns within community pediatric settings. In the current work, we significantly improve upon and expand Autoscreen based on usability data and interview data collected in a pilot investigation of pediatric healthcare providers using Autoscreen. The enhanced version of Autoscreen addresses limitations of the previous tool, such as scalability, and introduces important new features based on rigorous interviews with the target user population. Once validated on a large sample, Autoscreen could become an impactful tool for early ASD screening and targeted referral in primary care settings. The comprehensively-enhanced tool described in the current work will enable the investigative team to achieve this goal.

Keywords: Autism Spectrum Disorders · Digital screening · Usability enhancement · Scalability

1 Introduction

Autism Spectrum Disorders (ASD) are classified in the Diagnostic and Statistical Manual of Mental Disorders, 5th Edition (DSM-5) as neurodevelopmental disorders marked by (a) deficits in social communication and interaction and (b) repetitive and restrictive patterns of behavior and interest [1]. ASD often negatively affects lifespan outcomes, especially with regards to meaningful social engagement and occupational attainment [2]. Moreover, ASD prevalence has been on the rise for over a decade and, according to the Centers for Disease Control and Prevention, is now at its highest ever rate of 1 in 59 among children in the United States [3]. Therefore, early accurate identification and treatment of young children with ASD represents a pressing public health and clinical care challenge. Given mounting evidence that early, accurate diagnosis of ASD is possible and that very young children who receive intervention can demonstrate substantial gains in functioning, current American Academy of Pediatrics practice guidelines endorse formal ASD screening at 18 and 24 months of age [4, 5]. Unfortunately, large numbers of children are still not screened for ASD, waits for specialized diagnostic assessment can be very long, and the average age of diagnosis in the US remains between 4 to 5 years of age [3].

Continuing our earlier work [6], the current project uses pilot test and interview data to rigorously enhance and extend *Autoscreen*—a digital tool for time-efficient screening, diagnostic triage, referral, and treatment engagement of young children with ASD concerns within community pediatric settings. While technically novel in many respects, *Autoscreen* is not the first digital application designed to streamline screening of ASD in toddlers in clinical settings. For instance, CHADIS (Child Health and Development Interactive System; chadis.com) [7] is a web-based platform for administering screeners and assessments (e.g., M-CHAT) to a variety of populations, such as children with ASD and adolescent females with eating disorders. Although CHADIS supports some instruments for ASD screening and assessment, only *Autoscreen* has the specific focus of ASD risk assessment in toddlers based on a very brief (i.e., approximately 15-min) actively-guided interaction with minimal training required for administration. *Cognoa* [8] is another related application that uses a mobile application to provide a variety of screening and assessment services for individuals with concerns related to ASD and Attention Deficit Hyperactivity Disorder among others (cognoa.com). *Cognoa*, intended for use by clinicians and parents, uses manually coded video analyses of children’s behavior to make risk predictions. However, *Cognoa* is not designed to facilitate highly rapid administration in constrained clinical settings, as it requires a video review process by a team of remote clinicians.

2 Pilot Investigation of Usability

2.1 Implementation

The technical design of *Autoscreen* was previously described in [6]. The application was designed using the Unity game engine, which, despite being convenient for rapid deployment across a wide range of platforms (e.g., iOS, Android, and desktop

operating systems), is not an ideal tool for designing resource-efficient and scalable mobile applications. However, given these limitations, we successfully implemented a fully-functional prototype which reliably and accurately presented the novel procedures and screener content to pediatric care providers. The prototype application incorporated (a) a guided interface for the interactive administration of the assessment procedures, (b) a built-in behavioral coding interface, (c) automatic report generation detailing assessment results (including both dichotomous and continuous scale risk indices as well as noted areas of concerns and identified strengths), and (d) a video model of Autoscreen's procedures carried out between a provider and a child with ASD concerns (see Fig. 1).



Fig. 1. Screen capture from the video model: the provider and child engage in the turn-taking activity while the tablet running Autoscreen is out of the view of the child.

Providers interacted with Autoscreen remotely using two wireless peripheral devices that connected to the testing tablet via Bluetooth: an earbud and a presentation remote. This approach allowed providers to interact with the application without drawing the child's attention to the presence of the tablet. Providers progressed through test items using the wireless presentation remote which featured buttons for convenient navigation (i.e., forward/backward through test items), timer activation, and audio prompt activation. With the wireless earbud, providers were also given real-time audio instructions by the application, thus allowing providers to obtain testing instructions without distracting the child. The application provided clear, short, and simple instructions to lessen initial training burden, for which a short pre-administration tutorial was embedded into the application. Scoring was completed within the application via prompts which guided the provider to rate the child's responses in easy-to-report Likert categories (often, inconsistent, rare/not observed).

2.2 Usability Testing

In our IRB-approved pilot investigation, $n = 32$ professionals and paraprofessionals licensed to conduct ASD evaluations (e.g., developmental-behavioral pediatricians, clinical psychologists, speech-language pathologists) were recruited to assess the usability and preliminary validity of Autoscreen. Participants also included $n = 32$ families, i.e., a parent/caregiver and a child (18–36 months of age) with clinically-verified ASD diagnosis. The study was designed to capture data that would allow the investigative team to identify areas in which to enhance usability and extend functionality to include user-requested features. Data collected from participants included responses to (a) the System Usability Scale (SUS) [9], (b) Acceptability, Likely Effectiveness, Feasibility, and Appropriateness Questionnaire (ALFA-Q) [6], and (c) semi-structured Customer Discovery style interviews [10].

Each session lasted approximately one hour (excluding consent/assent). This hour consisted of 20 min to introduce the provider to the novel tool and its associated screener, 20 min for the core Autoscreen procedures, and a 20 min interview with the provider to complete questionnaires and discuss usability issues.

2.3 Results and Discussion

Providers reported high levels of both usability (mean SUS = 86.93) and acceptability (mean ALFA-Q = 87.50) of Autoscreen. This level SUS is regarded as “excellent” in the literature [11], and the overwhelming majority of providers reported “good” to “excellent” usability or better (see Table 1). The most common issue affecting usability reported by providers concerned minor difficulties involving the wireless earpiece through which audio prompts were given. As a result, we have transitioned to a new earpiece design that securely wraps around the participant’s ear.

Table 1. Overview of participant-reported scores on the system usability scale

Adjective	SUS cutoff	% Respondents above cutoff
Worst imaginable	12.5	0%
Awful	20.3	0%
Poor	35.7	0%
OK	50.9	6%
Good	71.4	25%
Excellent	85.5	25%
Best imaginable	90	44%

SUS = System usability scale [9]; See [11] for interpretation of scores

We next sought to identify—through Customer Discovery interviews [10]—updates and features viewed as highly desirable by participants that should be included in the next version of Autoscreen. We discovered three important results. First, we were able to clearly identify a set of meaningful value propositions based on reported clinical

care challenges. Interviewees were unanimously positive about the potential of Autoscreen and the vast majority indicated a strong desire to use the tool within their practices. Second, we identified participant-requested features that could provide substantial additional value (e.g., a digital note-taking feature to quickly record behavioral notes on-the-fly). Third, interviewees provided feedback about the broader context into which Autoscreen might ultimately be deployed, as well as how such a system could be optimally integrated within existing systems of care. For example, some interviewees suggested that Autoscreen could function well not just in face-to-face evaluations, but in remote administration scenarios as well (i.e., tele-medicine services for rural families). Additionally, interviewees reported numerous suggestions about aesthetic changes, many of which were incorporated and are discussed in Sect. 3.

3 System Enhancement and Extension

3.1 Frontend

Concurrent with the pilot study in which we collected user feedback about usability and satisfaction with the Autoscreen prototype, we commenced work on the enhanced application. Whereas the prototype was created in the Unity game engine, subsequent development of Autoscreen was performed using Android Studio. At present, Autoscreen can run on both tablets and phones, although the visual content is optimized for tablets 8" or larger. With regards to aesthetic changes, the revised color-palette was selected to achieve optimal readability and legibility [12]; the application has a logo in white and light green, on a backdrop of blue and cyan and now features a consistent visual template (Fig. 2). As requested by providers, the timer screen now features a larger font that is much easier to see at a distance (Fig. 2). After specified intervals, an audio clip is played indicating the time remaining for a particular task. In addition, buttons for timer play/pause and reset are available for use as needed.

Just as in the prototype application, the screening process begins with a form for subject data entry, and metadata such as date and time are automatically captured. Here, the provider enters the subject's anonymized identifier as well as the date of birth using large drop-down menus. The forward and backward arrow buttons permit users to navigate through the Autoscreen application screens, or "Activities" in android terms, and the *onBackPressed()* method has been overridden to switch back to the previous Activity using the physical back button on the tablet. Unlike the prototype application, the revised layout is now consistent across the entire application (see side-by-side comparison in Fig. 2). The home button located in the top left corner of the screen permits the user to pause the current activity in order to restart or go to review tutorial information. This functionality was not available in the Autoscreen prototype. A progress bar is located on the left side of the screen and displays five distinct sections of the application: subject data entry, materials checklist, primary screening activities, post-procedures screener, and risk profile report. The application header indicates at all times the section that the provider is currently in. Each page contains an audio tip button that allows the user to receive specific instruction for that page by clicking the button.

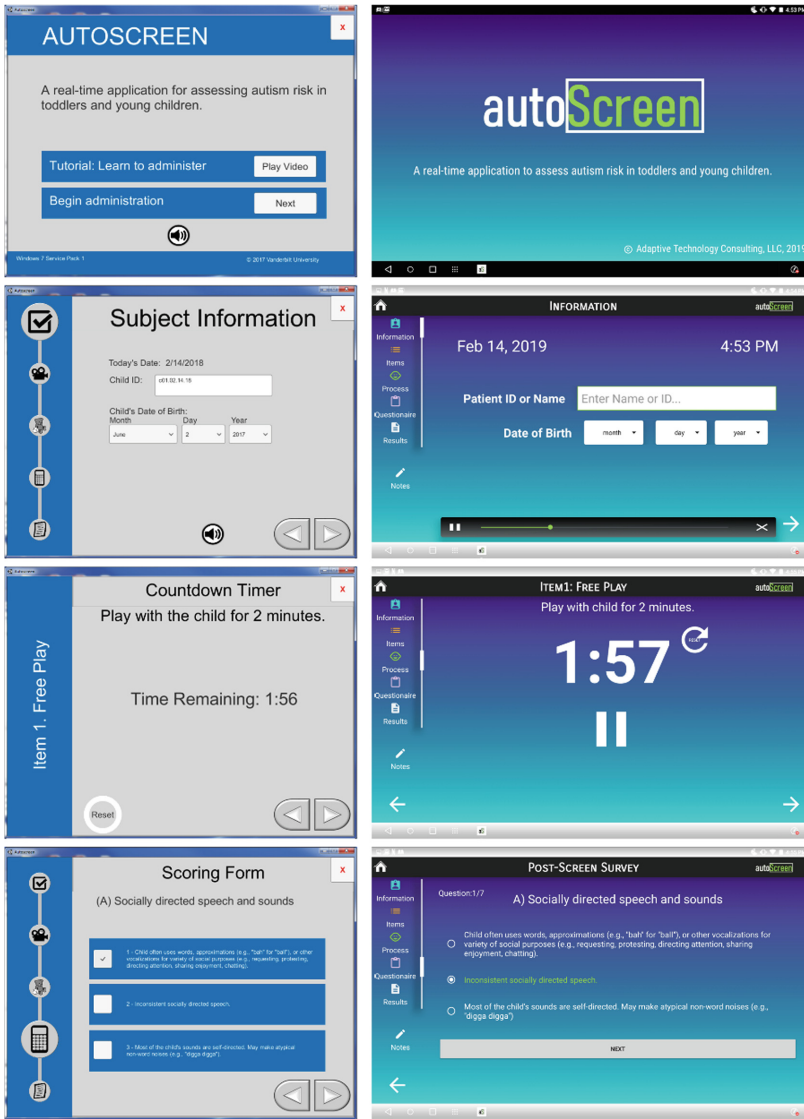


Fig. 2. Side-by-side comparison of the Autoscreen prototype (left column) and the enhanced implementation (right column).

The previously produced video model can now be streamed directly into the application and displayed in a media player. When a video is played, a small media player (Fig. 3) appears and includes functions for play, pause, and stop, and additionally features an interactive progress bar.

A closed caption button was also added to the media player, accommodating users who may prefer to read the text in addition to receiving the audio instructions. Subtitle

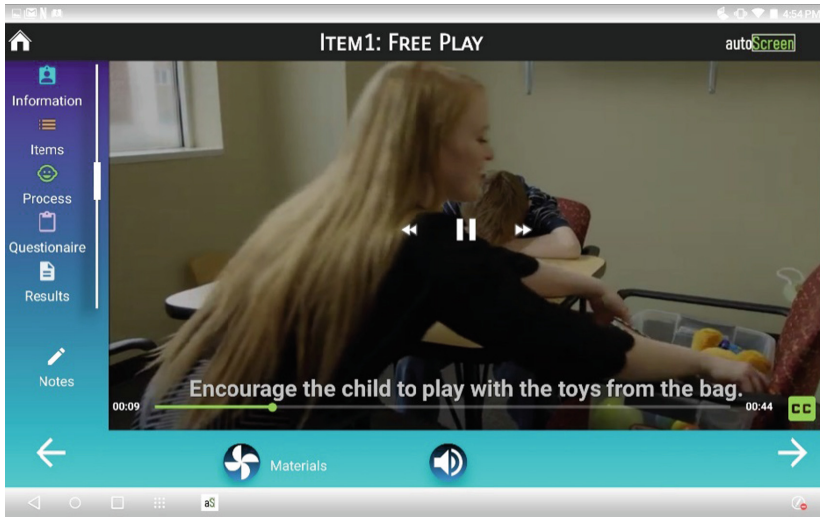


Fig. 3. Screen capture showing the newly introduced media player and audio subtitle features of Autoscreen.

text were loaded from *.srt* (SubRip Text) files. Below the media player, there is an additional button labeled “Materials”. Tapping on this button produces a pop-up dialog showing pictures of the items required for that particular stage of screening. The dialog box is an unobtrusive feature as it does not interrupt the video currently playing. To minimize the dialog, the user can touch the screen anywhere around the dialog box.

Many providers from the pilot study also requested a feature to facilitate on-the-fly note-taking, which has now been added to the current application. The note-taking widget is a separate layout with a multiline *EditText* box. The user can type notes on any page that they like and at any time, and the save button will store the text in a file associated with the session (see Fig. 4).

3.2 Backend

The current work utilizes Amazon Web Services to host and facilitate a centralized backend model. Enhancements in the current backend model now include the ability to handle requests from frontend clients. The backend is segregated into (a) a REST API endpoint server based on Flask [13], and (b) a predictive model utilizing Numpy (www.numpy.org) for data-processing, and Keras (www.keras.io) for a model-based prediction. The development stack was built entirely in Python 3.6 and included Flask for its straightforward approach to developing functional, full-featured RESTful API endpoints, and Keras for high-level abstraction to design, train, test, and deploy neural networks. Additionally, Keras uses the default TensorFlow backend for its lower-level component and GPU access when available. Numpy handles common data manipulation and is requisite for most data-science applications. Multithreaded applications such as Keras, TensorFlow, and even Python in general are especially problematic for

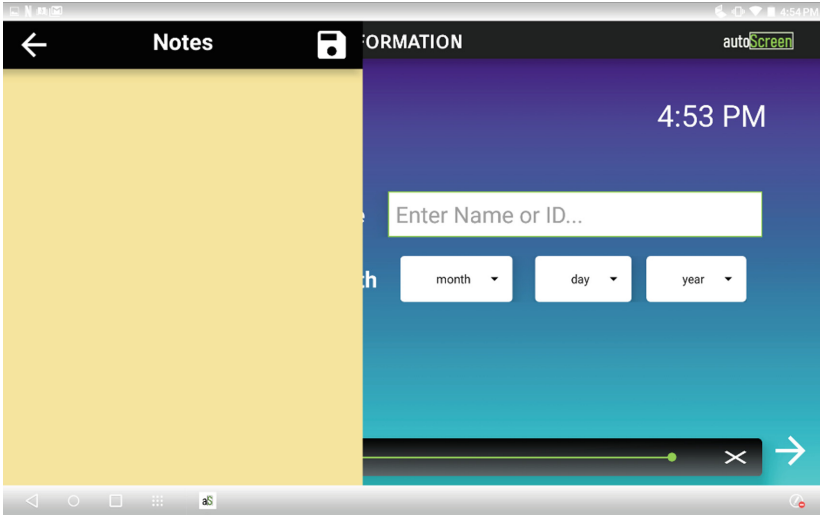


Fig. 4. Newly added feature for on-the-fly note-taking.

asynchronous, event-driven, context-sensitive frameworks such as Flask. As a result, Autoscreen’s server splits these into two closely matched, socket-connected daemon applications. Python’s *SocketServer* from the standard library provides a high-level abstraction for OS-level system calls to connect each process.

This design has several advantages apart from overcoming some multithreading challenges. First, the system is modular as the API endpoints are not directly tied to any specific predictive model, thus permitting models to be changed or updated over time; Flask interacts with a *SocketServer* and passes parameterized commands for some model to operate on (e.g., to make predictions on batch data). Both platforms must agree on an internal argument or API. In fact, the API-facing interface contains endpoints to receive an entire pre-trained model and can be instructed to load a model from an HDF5 file. Future work will implement functionality for sending persistent or “pickled” models. This design will allow the predictive model and API handler to exist on different machines. The API handler could theoretically “round-robin” instructions to predictors on different machines for load balancing via socket connections. Future work will introduce the flexibility to test experimental models and to ensure accountability as models change over time.

Finally, in order to validate the backend architecture’s ability to handle client requests, work continued on the development of a predictive model based on a subset of test items used to recognize ASD. A sequential, feed-forward, deep-net makes predictions at a current accuracy of upwards of 95% based on 90/10 testing split for 739 examples, which is based on the exploratory dataset described in our earlier work [6]. The reader should note that this exploratory model does not represent Autoscreen-generated features at this time, and thus is intended as a benchmark for an idealistic upper-bound on the potential real-world performance of Autoscreen. In lieu of a model validated explicitly on Autoscreen data, however, the functional backend architecture

represents a major step forward in advancing Autoscreen to the level of scalable deployment. In sum, initial development on hosting servers, installation and configuration of a running environment, and DNS record-keeping have established a robust publicly-accessible interface. Future work will seek to establish authorization mechanisms as well as trust relationships.

4 Conclusions

The current work builds significantly on our previous work, making several new contributions. First, our current frontend was redesigned based on extensive feedback from the target user audience of pediatric care providers who assess risk for ASD. This included both an overall aesthetic update and the addition of many new user-requested features. Second, although our earlier prototype was fully-functional, it lacked the ability to scale given the choice of development environment (i.e., the Unity game engine). Now, our heavy investment in the new backend architecture will facilitate scalable client-server communication for dynamic risk profile generation. While the current data sample collected from tests of Autoscreen is too small to permit real-world deployment at this time, our current system represents a significant step towards achieving that goal. Future research and development will involve continual improvement of both the user interface and the server-side components of Autoscreen, as well as a large-scale study to capture data from a sample sufficiently large for validation of the novel tool.

Acknowledgement. This work was supported by the National Institute of Mental Health of the National Institutes of Health through award number 1R43MH115528. The authors would like to express thanks to the families and pediatric care providers who gave significant time for participation in this study.

References

1. American Psychiatric Association: Diagnostic and statistical manual of mental disorders (DSM-5®). American Psychiatric Publishing (2013). <https://doi.org/10.1176/appi.books.9780890425596>
2. Howlin, P., Goode, S., Hutton, J., Rutter, M.: Adult outcome for children with autism. *J. Child Psychol. Psychiatry* **45**(2), 212–229 (2004). <https://doi.org/10.1111/j.1469-7610.2004.00215.x>
3. Baio, J., et al.: Prevalence of autism spectrum disorder among children aged 8 years — autism and developmental disabilities monitoring network, 11 sites, United States, 2014. *MMWR Surveill. Summ.* **67**(6), 1–23 (2018). <https://doi.org/10.15585/mmwr.ss6513a1>
4. Dawson, G., et al.: Randomized, controlled trial of an intervention for toddlers with autism: the early start denver model. *Pediatrics* **125**(1), e17–e23 (2010). <https://doi.org/10.1542/peds.2009-0958>
5. Warren, Z., McPheeters, M.L., Sathe, N., Foss-Feig, J.H., Glasser, A., Veenstra-VanderWeele, J.: A systematic review of early intensive intervention for autism spectrum disorders. *Pediatrics* **127**(5), e1303–e1311 (2011). <https://doi.org/10.1542/peds.2011-0426>

6. Sarkar, A., Wade, J., Swanson, A., Weitlauf, A., Warren, Z., Sarkar, N.: A data-driven mobile application for efficient, engaging, and accurate screening of ASD in toddlers. In: Antona, M., Stephanidis, C. (eds.) UAHCI 2018. LNCS, vol. 10907, pp. 560–570. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-92049-8_41
7. Sturmer, R., Howard, B., Bergmann, P., Stewart, L., Afarian, T.E.: Comparison of autism screening in younger and older toddlers. *J. Autism Dev. Disord.* **47**(10), 3180–3188 (2017). <https://doi.org/10.1007/s10803-017-3230-1>
8. Kanne, S.M., Carpenter, L.A., Warren, Z.: Screening in toddlers and preschoolers at risk for autism spectrum disorder: evaluating a novel mobile-health screening tool. *Autism Res.* **11**(7), 1038–1049 (2018). <https://doi.org/10.1002/aur.1959>
9. Brooke, J.: SUS—a quick and dirty usability scale. *Usability Eval. Ind.* **189**(194), 4–7 (1996). <https://doi.org/10.1201/9781498710411>
10. Blank, S., Dorf, B.: *The Startup Owners Manual*, vol. 1. K&S Ranch Inc., Pescadero (2012)
11. Bangor, A., Kortum, P.T., Miller, J.T.: An empirical evaluation of the system usability scale. *Int. J. Hum.-Comput. Interact.* **24**(6), 574–594 (2008). <https://doi.org/10.1080/10447310802205776>
12. Hall, R.H., Hanna, P.: The impact of web page text-background colour combinations on readability, retention, aesthetics and behavioural intention. *Behav. Inf. Technol.* **23**(3), 183–195 (2004). <https://doi.org/10.1080/01449290410001669932>
13. Ronacher, A.: *Welcome—flask (a python microframework)* (2010). <http://flask.pocoo.org/>. Accessed 30 Jan 2019