Pennsylvania State University

College of Information Sciences and Technology

IST 521 – Fall 2020

Doxector: A Web Application for Detecting Doxing in Twitter

Jennifer Farnum and Younes Karimi

{jjf5499, younes}@psu.edu

December 15, 2020

Abstract

Doxing is a common phenomenon in microblogs and social media websites such as Twitter, in which people disclose sensitive information about others without their consent. In this project, we aim to create a web application that takes new tweets as input through an interface and returns an assessment on whether the tweet is potentially doxing (or generally, disclosing sensitive information about others) or not. The application can be used by individuals or enforced by Twitter on its new tweets to avoid unintended private information disclosure. To this end, we create a supervised learner by first preprocessing and vectorizing each tweet using the existing natural language processing tools and then building a model based on our samples for doxing and non-doxing tweets. The presented application interface is aimed to take new tweets from users and send them to our automated system to vectorize and compare the tweets against the model we have built with the known samples. The interface also provides sign-up and login capabilities so that the users can see their history of requested URLS. To enhance our design for the application interface, we perform web accessibility and usability evaluations and improve our low-fidelity prototype.

Contents

1 Introduction

Twitter is a popular social media application that has over 330 million users. Their privacy policy and privacy team often react to what is known as doxed information. *Dox* is an abbreviation for *documents* and the term *Doxing* refers to collecting PII, sensitive and private information (i.e., documents or doxed information) about others and disclosing them publicly without their consent (Douglas, 2016) Doxed information is publicly broadcasting private or personal identifiable information. This nonconsensual disclosure may threat people's jobs, families, or even their lives (Bellmore, Calvin, Xu, & Zhu, 2015; Chen, Chan, & Cheung, 2018). It may also cause defamation of public figures or celebrities (Basak, Sural, Ganguly, & Ghosh, 2019).

Doxed information is often the result of reverse engineering, hacking, or other malicious forms of soliciting information from users. A tweet may be considered as doxing if:

- The tweet's content reveals some sensitive or private information. Note that depending on the context, some information may be considered sensitive or not sensitive. For instance, since a company's support phone number is generally publicly available and intended to be public, it should not be considered as sensitive. But, disclosing a phone number that belongs to an individual or is being used personally can be considered as a doxing action.

- This sensitive information is about a second or third party (without their consent) and not about the authors' themselves (not a self-disclosure)

Although plenty of doxing tweets can be identified easily, there can be some controversial ones in which making an appropriate decision may require additional contextual information. This information can be the target of the potentially doxing information, whether the tweet is replied to or mentions another Twitter account, other tweets of the author, etc. Therefore, deciding based on a single tweet can be subjective and dependable on the subject who annotates the tweets, their personal characteristics, knowledge, and thoughts.

While doxing may happen in many different social networking platforms such as forums, blogs, chat rooms, or websites, we have specifically focused on Twitter. This is because Twitter has a unique data structure and metadata, which allows us to perform a more in-depth analysis of the tweets. Twitter

also provides a public REST API for collecting tweets. Due to its structure and usage, we presumed people dox others and share sensitive information about others more often on Twitter.

In this lab, we aim to use some basics presented by (Ericsson & Simon, 1980, 1984, 1993) as a model of how subjects think aloud and verbalize the information they are thinking of in response to a set of instructions.

2 Methods

A task analysis was performed using an automated accessibility tool. The Web Accessibility Evaluation Tool (WAVE) was chosen because of its unique ability to evaluate live websites for accessibility errors, so people of all abilities can use a given website without issue. WAVE can identify many accessibility and Web Content Accessibility Guideline (WCAG) errors and facilitate the human evaluation of web content.

The WAVE tool was chosen because of its availability-WAVE is a free tool available as a browser extension, so a given user can quickly run it on any web page. WAVE shows errors by identifying accessibility errors using red icons, including missing alt text, empty links and missing headings, accessibility warnings as yellow icons, accessible elements already in place as green icons, areas of low color contrast, and provide information about what errors mean and how to fix them.

This tool was first envisioned by Dr. Len Kasday at Temple University in 2001 and has since become the industry standard for identifying errors and facilitating human evaluation.

2.1 Materials

The materials used include:

- Laptop running macOS 10.14 (2880 x 1800 resolution)
- Adobe XD
- WAVE Web Accessibility Evaluation tool[1]
- Kaltura Capture

[1]https://wave.webaim.org

2.2 Doxector Application Prototype

The term *Doxector* is generated by combining the two terms, *Dox* and *Detector*, and Doxector is a web application prototype for detecting doxing in Twitter.

The application prototype can detect sensitive information disclosure in tweets and identifies whether a tweet is doxing or not. To that aim, it gets the URL to a public (unprotected) tweet and returns the results of its analysis on the tweet The analysis was performed using Twitter REST API, NLP, and ML techniques in the back-end.

Using Adobe XD and a laptop, both the low and high-fidelity prototypes were created using the Adobe program. The logo for the prototype was also created using Adobe XD.

The prototype provides the functionality for users to log in, sign up, or use the application as a guest. If a subject decides to use the application as a guest, they will have limited functionality. Subjects who use the application as a guest can only run a suspected doxing tweet through the program to see if it is constituted as doxing.

Subjects can log in using their email or Twitter account. If users log in to the application, they can run a suspected doxing tweet through the program to see if it constituted as doxing. Additionally, signed in users can see previously reported tweets.

The application allows for users to sign out after they are done using the application.

2.3 Doxector Demo

To better illustrate the functionality of our designed interface for Doxector, we have created a walk-through video demo showing its different screens, messages, and capabilities. The video was created using Penn State's Kaltura[2] program, a Zoom extension.

2.4 Design and Procedure

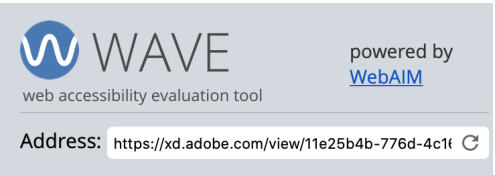
We used the WAVE tool to analyze our prototype to identify any structural elements and any existing or potential errors associated. Figure 2 shows the prototype homepage.

[2]https://bit.ly/Doxtector

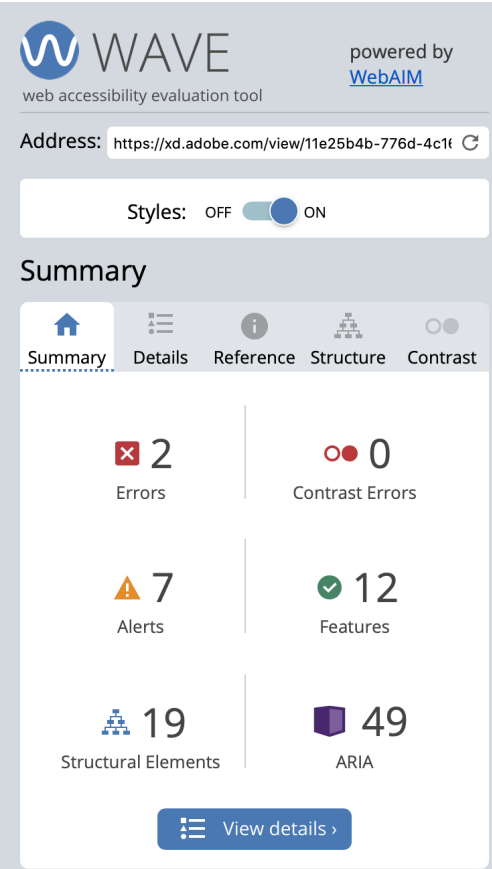
Figure 2

The subject entered the URL in the provided box and enabled the styles so that we can see the stylistic elements and visual representations and images used in the web page (Figure 3a).

For each element or error, there is an item associated to it that may show further information and the code associated to that part or error can be identified from there. The tool also provides references to get more information about the errors an example of which can be seen in the Figure 3b. This error is about low contrast between colors of different segments in the page and the reference gives hints and suggestions for resolving the issue as well.



(a)



(b)

Figure 3: Prototype alt-boards

3 Analyses and Results

To enhance the performance and usability of our prototype, comply with accessibility standards, we ran different analyses.

3.1 Web Accessibility Evaluation

To make our interface user friendly and accessibly for people with disabilities, we performed a Web Accessibility Evaluation (WAE) based on the assessments we had already done during the semester in one of the labs, Au-

tomatic Testing (Bobby) Lab[3]. In that lab, we had used the *WAVE* Web Accessibility Evaluation Tool[4] from WebAIM[5] on our learning management website, Canvas[6] to identify the accessibility errors and warnings. While there are several more recent studies proposing new approaches and tools and comparing them against WAVE, we chose WAVE because it is a well-known, popular, and widely-used online and free tool that was introduced in this course and we had also already seen its performance in our course lab. It also provides a nice interface for testing web pages on local browsers. (Giovanna, Manca, Paternó, & Pulina, 2020) proposes a new approach called *MAUVE++* which supports *Web Content Accessibility Guidelines (WCAG) 2.1*[7], and they have shown that it has outperformed *WAVE* in terms of validity and completeness. (Alsaeedi, 2020) has proposed *SiteImprove*, an accessibility evaluation framework, and compared that with *WAVE* by evaluating several university homepages. (Padure & Pribeanu, 2020) compares six different evaluation tools and highlights the capability of *WAVE* in evaluating the contrast (which led to one of our prototype improvements), and non-styles content. Furthermore, (Roselli, 2020) has looked into the impacts of accessibility overlays on WAE tools such as *WAVE* and how they may spoof these automated tools. But, we did not use such overlays in our prototype.

3.1.1 WAVE

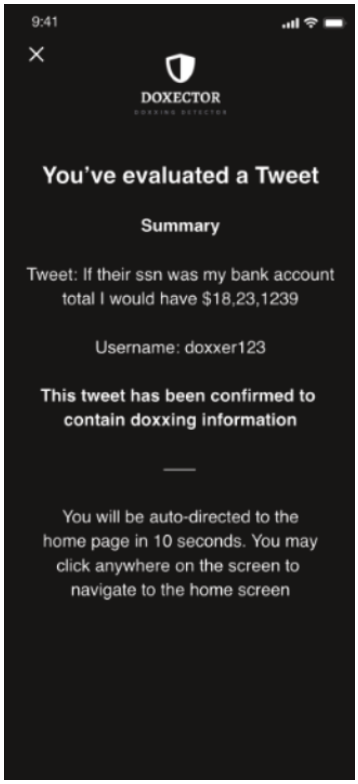
WAVE is a suite of evaluation tools that helps authors make their web content more accessible to individuals with disabilities. WAVE can identify many accessibility and Web Content Accessibility Guideline (WCAG) errors, but also facilitates human evaluation of web content. Our philosophy focuses on issues that we know impact end-users, facilitate human evaluation, and educate about web accessibility.

One can use the online WAVE tool by entering a web page address (URL) within the website. WAVE Firefox and Chrome extensions are available for testing accessibility directly within the web browser - handy for checking password-protected, locally stored, or highly dynamic pages.

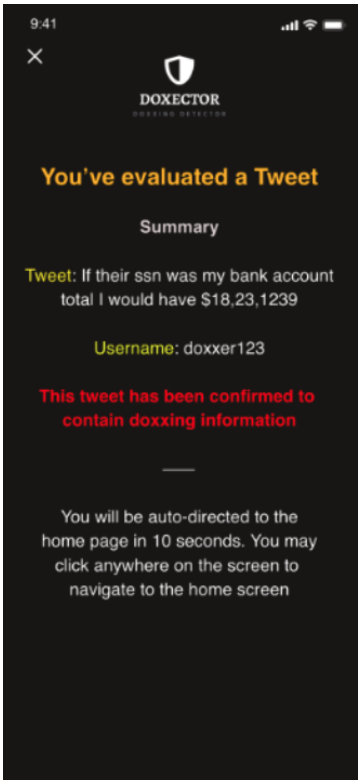
[3]http://acs.ist.psu.edu/ist521/bobby-lab2.txt
[4]https://wave.webaim.org
[5]https://webaim.org
[6]https://canvas.psu.edu
[7]https://www.w3.org/TR/WCAG21

3.1.2 Web Accessibility Enhancement

As it is shown in the Figures 4a and 5a, our interface initially did not have an appropriate color and font size contrast. Therefore, it would be harder for the users to differentiate between different sections in the interface windows. To resolve this issue, we used *Bold* and a different color for our keys and titles. The accessibility-enhanced windows are illustrated in the Figures 4b and 5b.

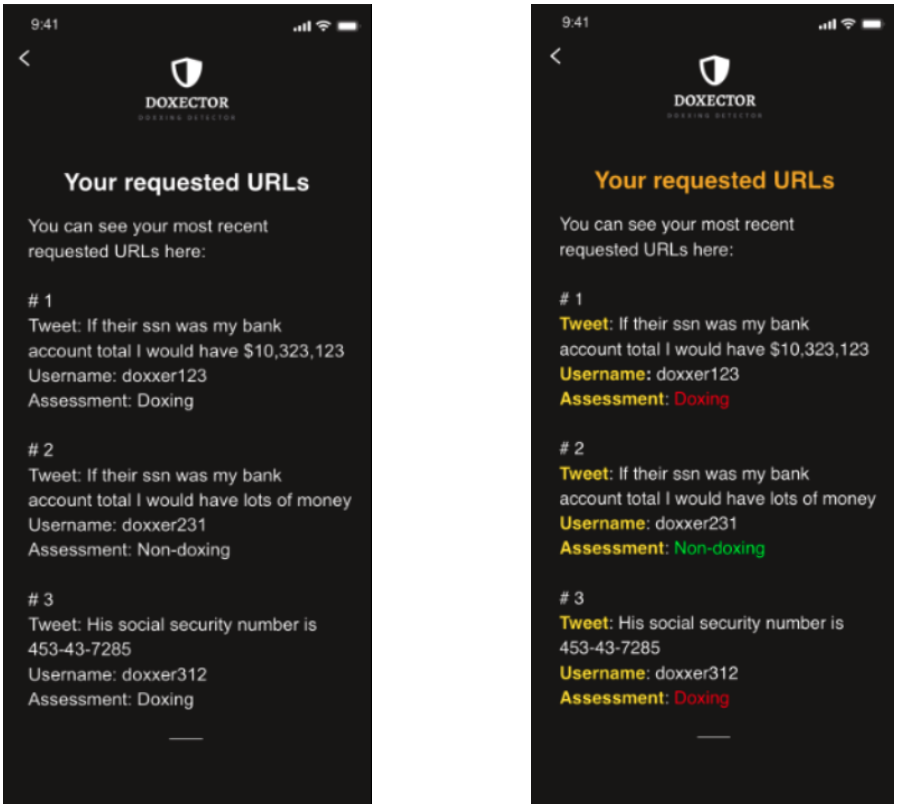


(a) First interface before accessibility enhancement



(b) First interface after accessibility enhancement

Figure 4: Interface enhancements based on web accessibility evaluations metrics



(a) Second interface before accessibility enhancement

(b) Second interface after accessibility enhancement

Figure 5: Interface enhancements based on web accessibility evaluations metrics

3.2 Usability Evaluation

As stated by (Pew, 2008), effective integration of human-system issues requires stakeholder satisfying. This means proposing solutions that meet acceptability criteria of all stakeholders. Moreover, it requires incremental growth of system definition and stakeholder commitment, iterative and concurrent system definition and development. Therefore, inspired by Spiral model (Boehm & Hansen, 2001), we presented an initial demo of our interface in the class and asked all our classmates and the instructor to provide feedback on that. They highlighted a missing story from our interface; if a user wants to use this application without having and creating any accounts

in the app and without the need to retrieve her requested URLs history. Therefore, we added a few more screens to our interface so that the user can use this app even without logging in.

4 Discussion and Conclusion

The WAVE tool was used to find accessibility errors within the Doxtector prototype. In summary, there were no critical contrast errors and seven warnings.

If developed upon further, the ability to test robust prototypes would help developers create an accessible product earlier on in the development life cycle. This would be a difficult task as there are various ways to create prototypes. However, a partnership with Adobe would be a step in the right direction. This would allow designers and developers of all skill levels to create accessible websites –creating meaningful progress in the field of human-computer interaction (Council et al., 2007).

The automatic tester allows an open-ended task analysis to be automated. This allows for an iteration of testing to be completed automatically before using human subjects. Concerning the spiral risk-driven model, this would allow developers to identify and resolve risks with respect to accessibility (Boehm & Hansen, 2001). An automatic tester removes the need for a subject to test a given website for accessibility. This enables web developers to create websites that cater to those who may fall under the vulnerable population category under IRB guidelines.

In further research, this prototype could be built out further to become a full-functioning mobile application or website with coordinating back-end functionalities. Additionally, this application could be used as a Twitter extension to confirm that a given tweet contains doxed information and then report the tweet to Twitter with the corresponding data to have it removed as soon as possible.

References

Alsaeedi, A. (2020). Comparing web accessibility evaluation tools and evaluating the accessibility of webpages: Proposed frameworks. *Information*, *11*(1), 40.

Basak, R., Sural, S., Ganguly, N., & Ghosh, S. K. (2019). Online public shaming on twitter: Detection, analysis, and mitigation. *IEEE Transactions on Computational Social Systems*, *6*(2), 208–220.

Bellmore, A., Calvin, A. J., Xu, J.-M., & Zhu, X. (2015). The five w's of "bullying" on twitter: Who, what, why, where, and when. *Computers in human behavior*, *44*, 305–314.

Boehm, B., & Hansen, W. J. (2001). The spiral model as a tool for evolutionary acquisition..

Chen, Q., Chan, K. L., & Cheung, A. S. Y. (2018). Doxing victimization and emotional problems among secondary school students in hong kong. *International journal of environmental research and public health*, *15*(12), 2665.

Council, N. R., et al. (2007). *Human-system integration in the system development process: A new look*. National Academies Press.

Douglas, D. M. (2016). Doxing: a conceptual analysis. *Ethics and information technology*, *18*(3), 199–210.

Ericsson, K. A., & Simon, H. A. (1980). Verbal reports as data. *Psychological review*, *87*(3), 215.

Ericsson, K. A., & Simon, H. A. (1984). *Protocol analysis: Verbal reports as data.* the MIT Press.

Ericsson, K. A., & Simon, H. A. (1993). Protocol analysis. verbal reports as data (revised edition). *MITP, Cambridge, MA*.

Giovanna, B., Manca, M., Paternó, F., & Pulina, F. (2020). Flexible automatic support for web accessibility validation. *Proceedings of the ACM on Human-Computer Interaction*, *4*.

Padure, M., & Pribeanu, C. (2020). Comparing six free accessibility evaluation tools. *Informatica Economica*, *24*(1).

Pew, R. W. (2008). Some new perspectives for introducing human-systems integration into the system development process. *Journal of Cognitive Engineering and Decision Making*, *2*(3), 165-180. Retrieved from `https://doi.org/10.1518/155534308X377063` doi: 10.1518/155534308X377063

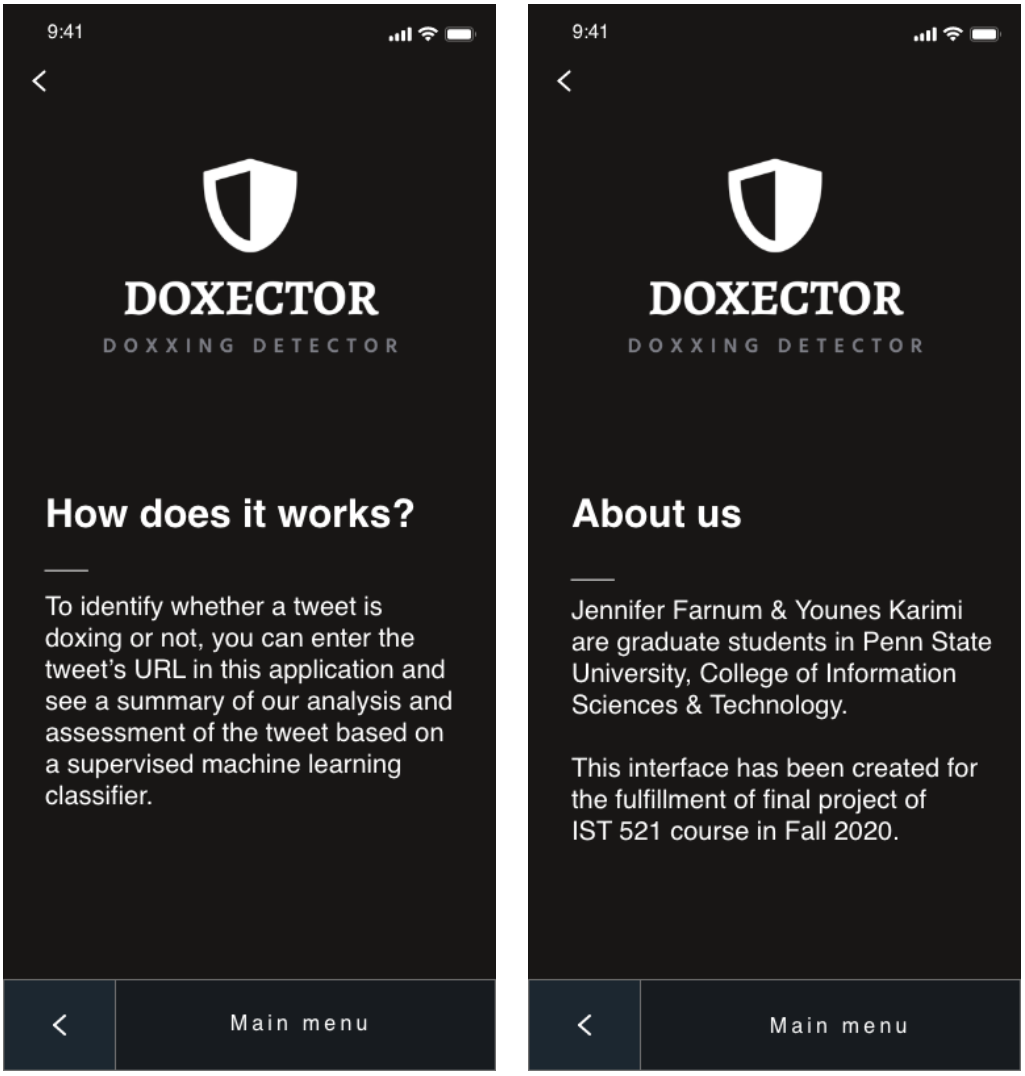
Roselli, A. (2020). *#accessiBe Will Get You Sued.* https://adrianroselli.com/2020/06/accessibe-will-get-you-sued.html.

Appendix

Doxtector Prototype

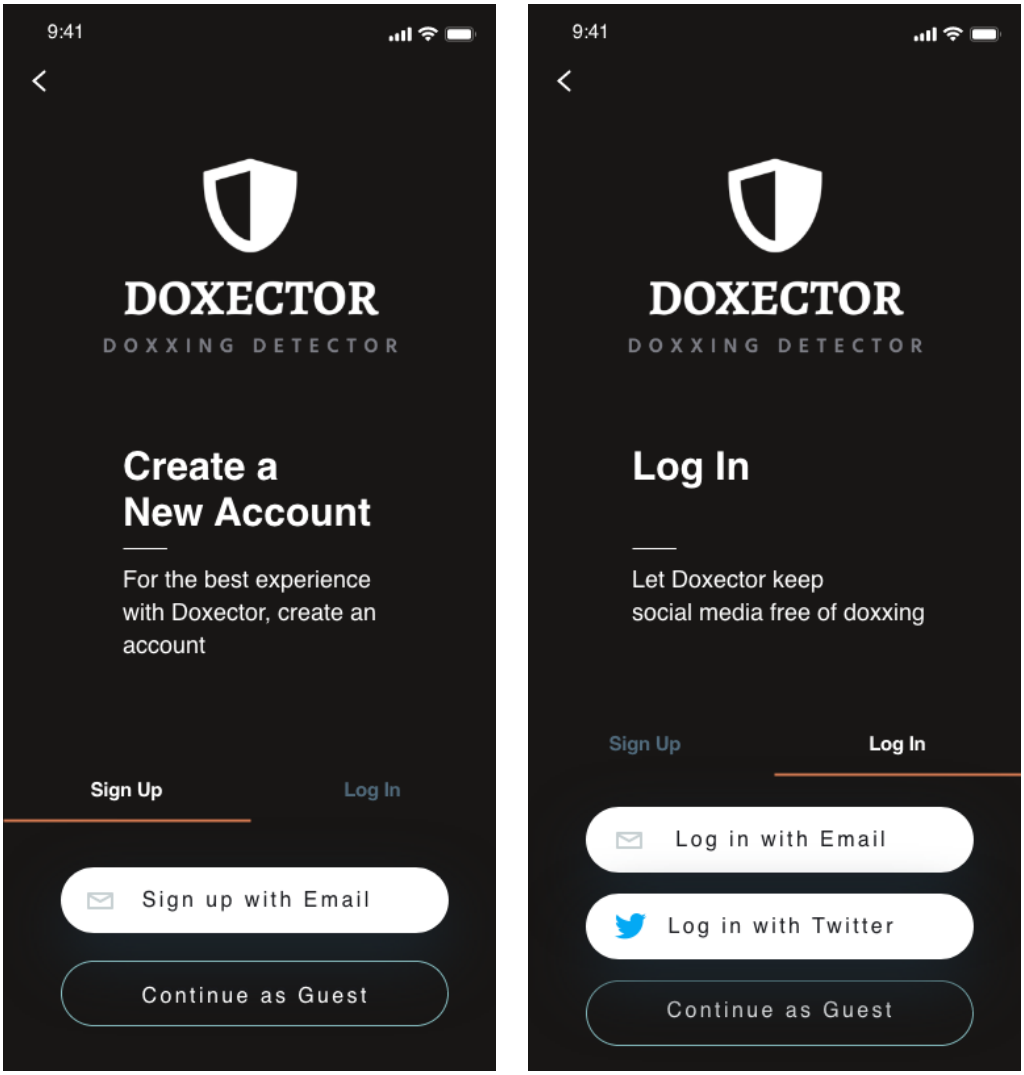
(a) (b)

Figure 6: Prototype alt-boards



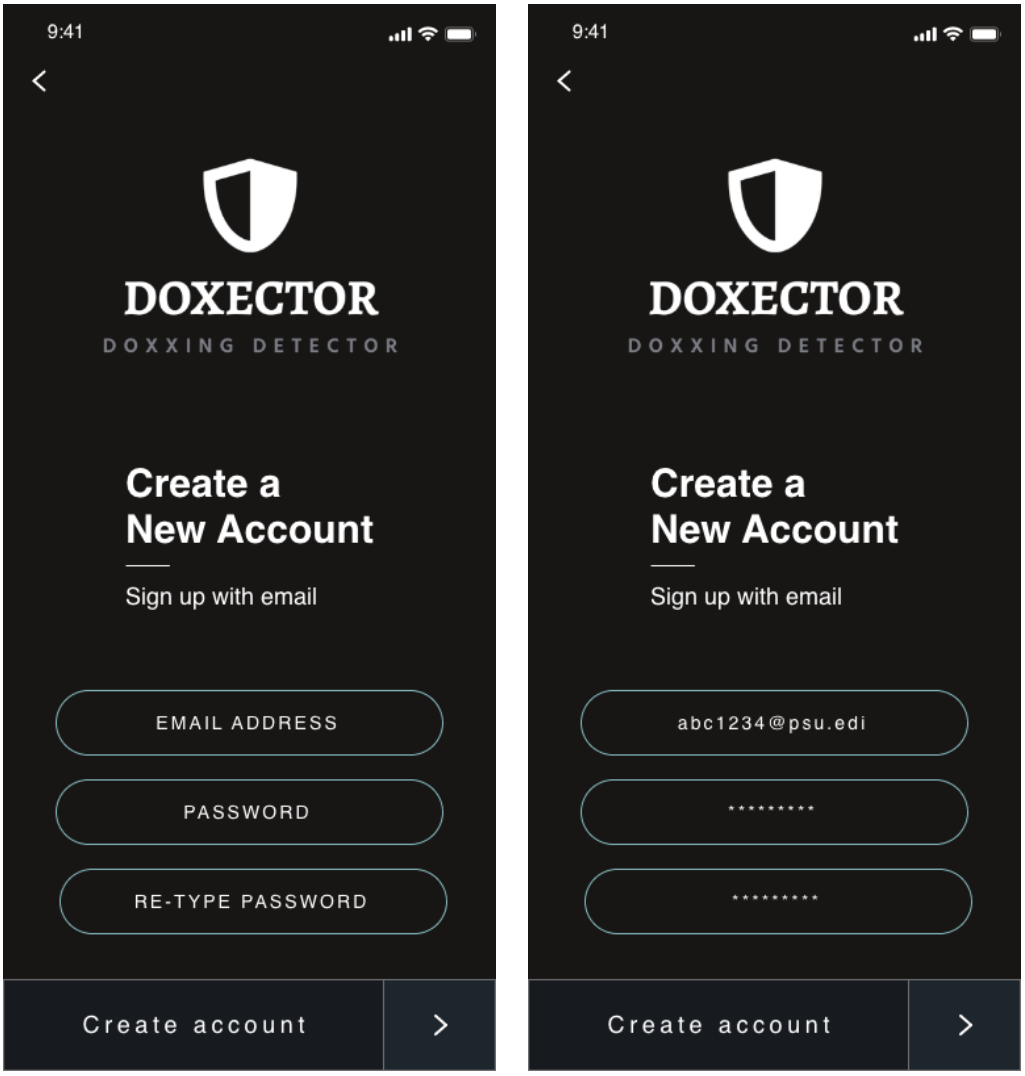
(a) (b)

Figure 7: Prototype alt-boards



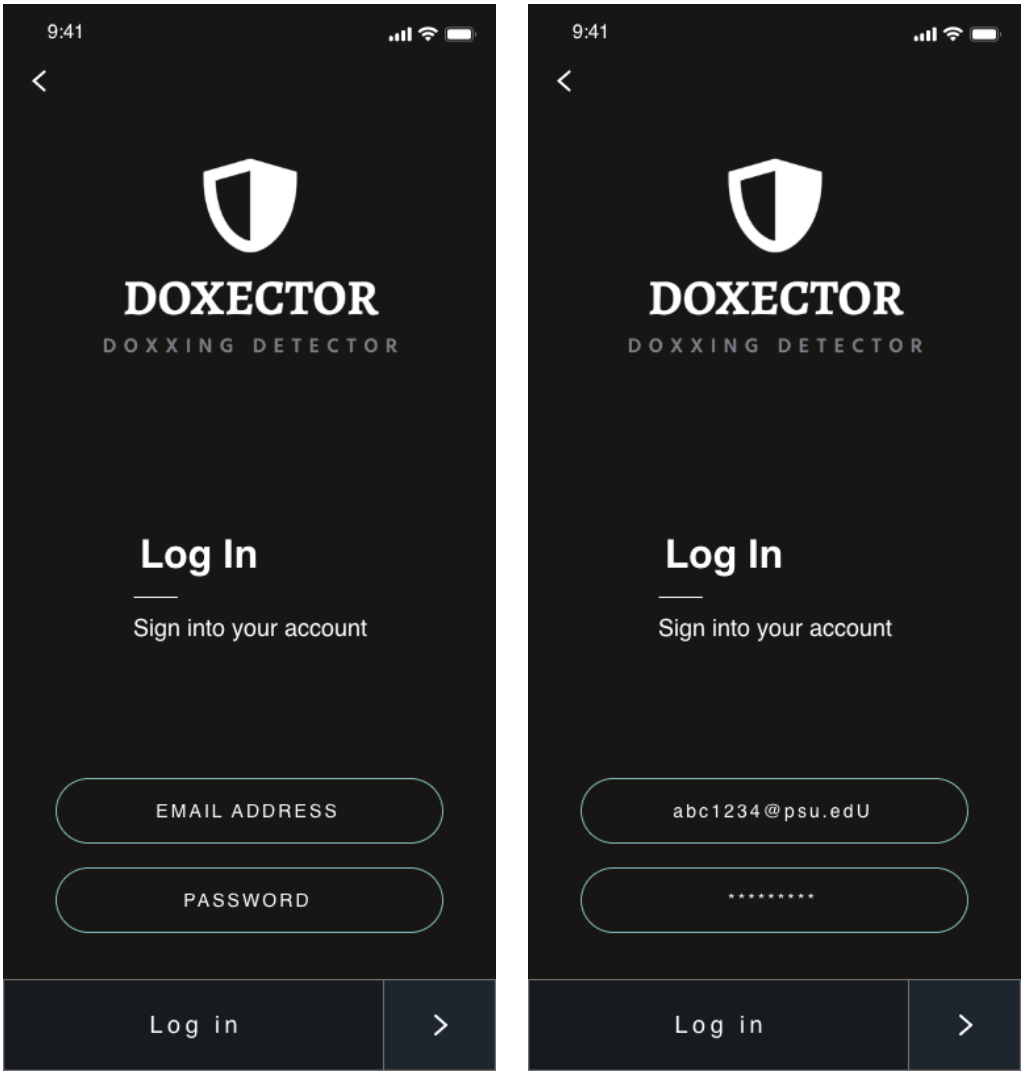
(a) (b)

Figure 8: Prototype alt-boards



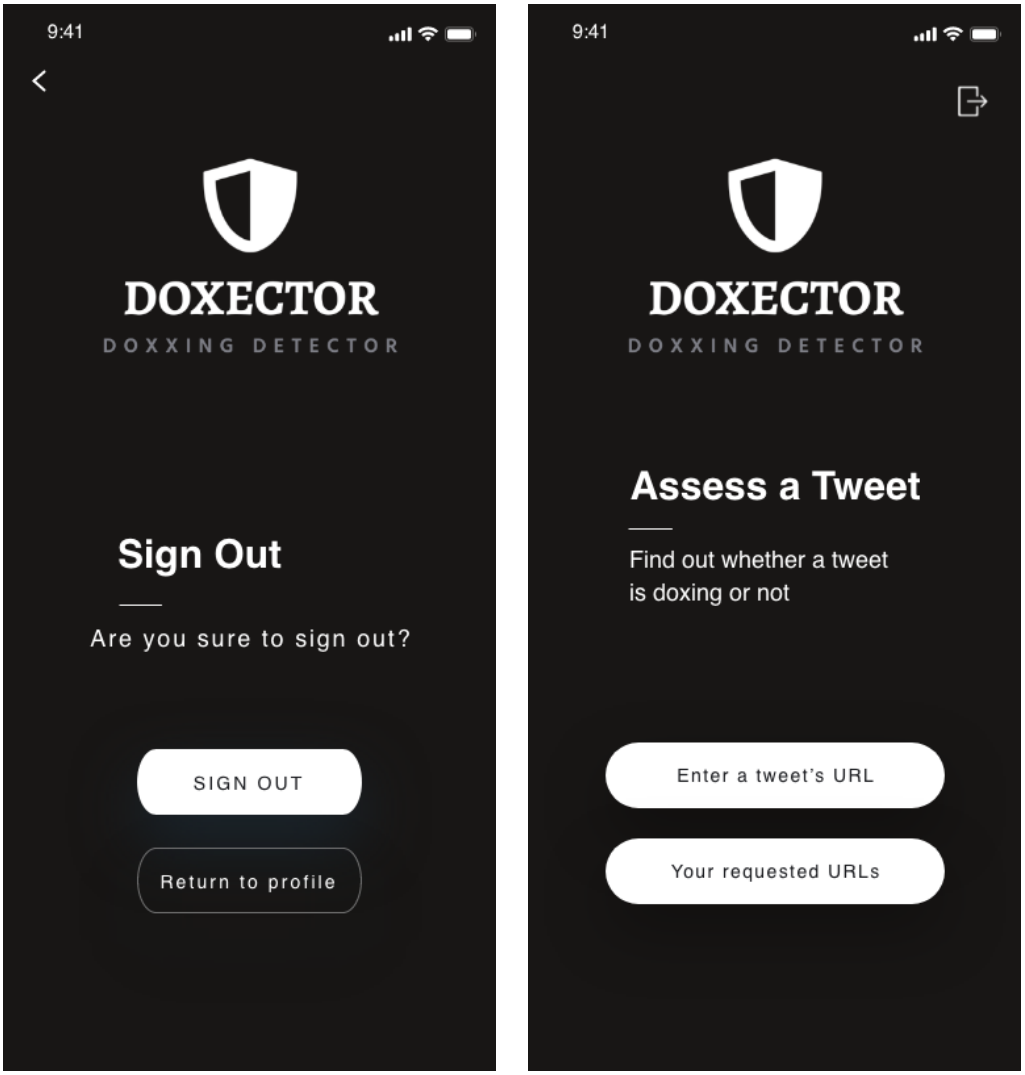
(a) (b)

Figure 9: Prototype alt-boards



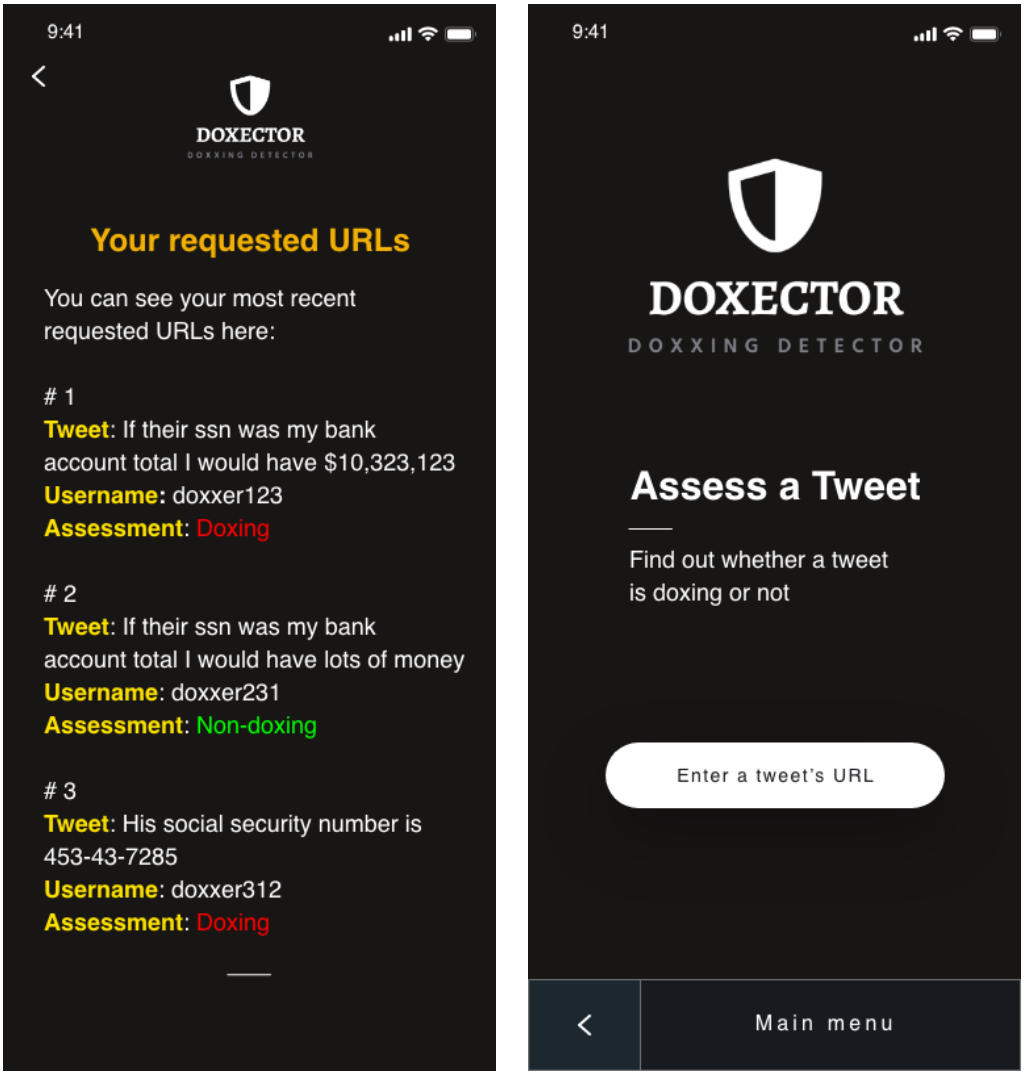
(a) (b)

Figure 10: Prototype alt-boards



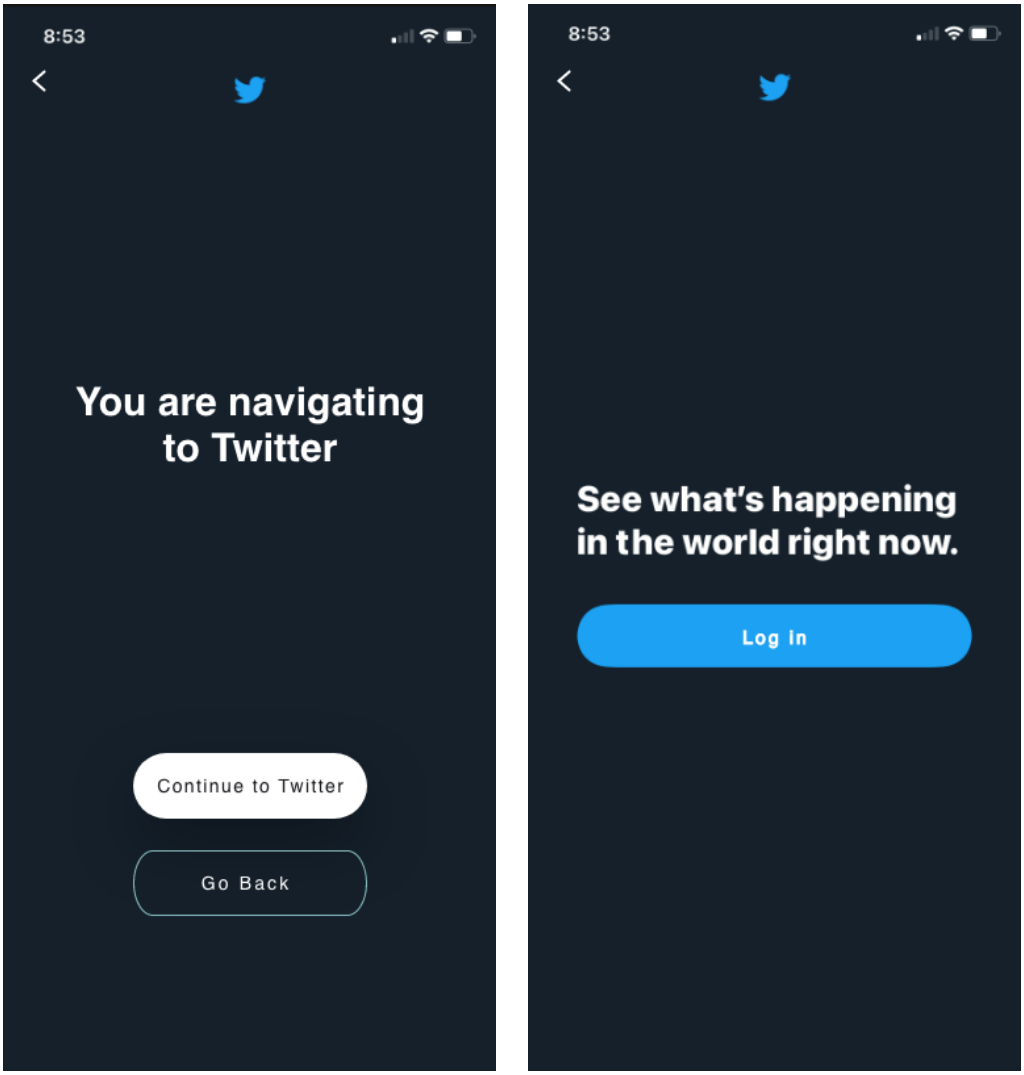
(a) (b)

Figure 11: Prototype alt-boards



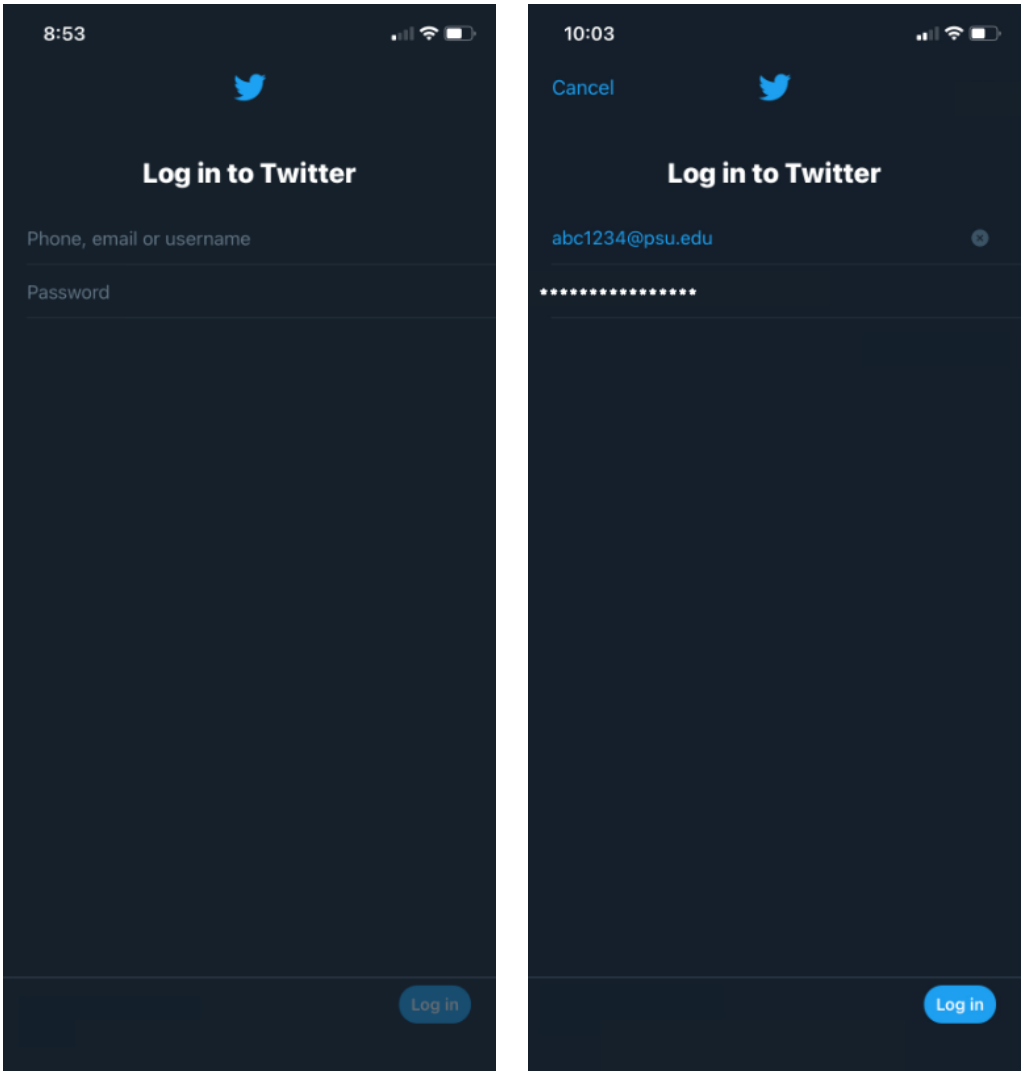
(a) (b)

Figure 12: Prototype alt-boards



(a) (b)

Figure 13: Prototype alt-boards



(a) (b)

Figure 14: Prototype alt-boards

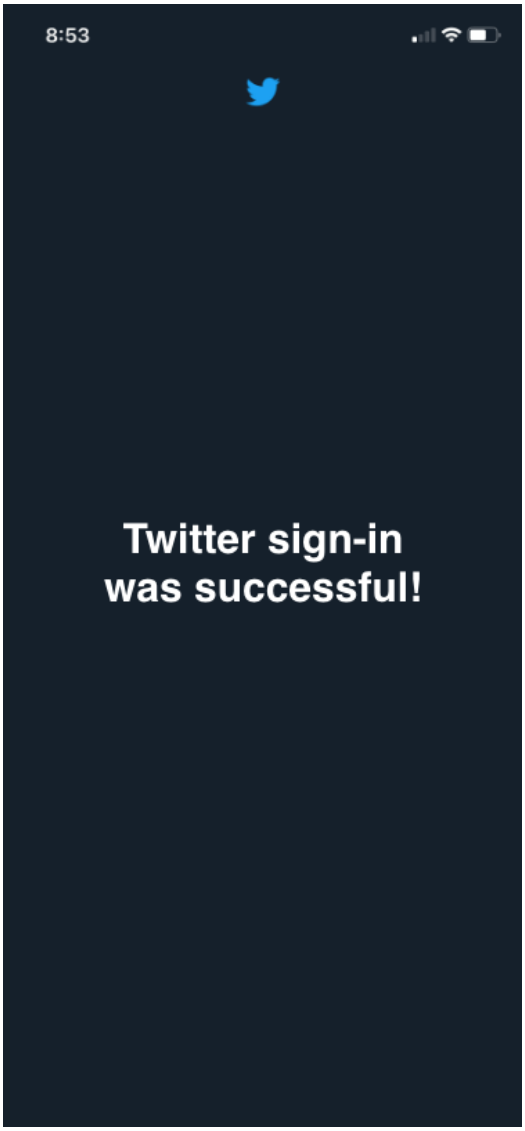
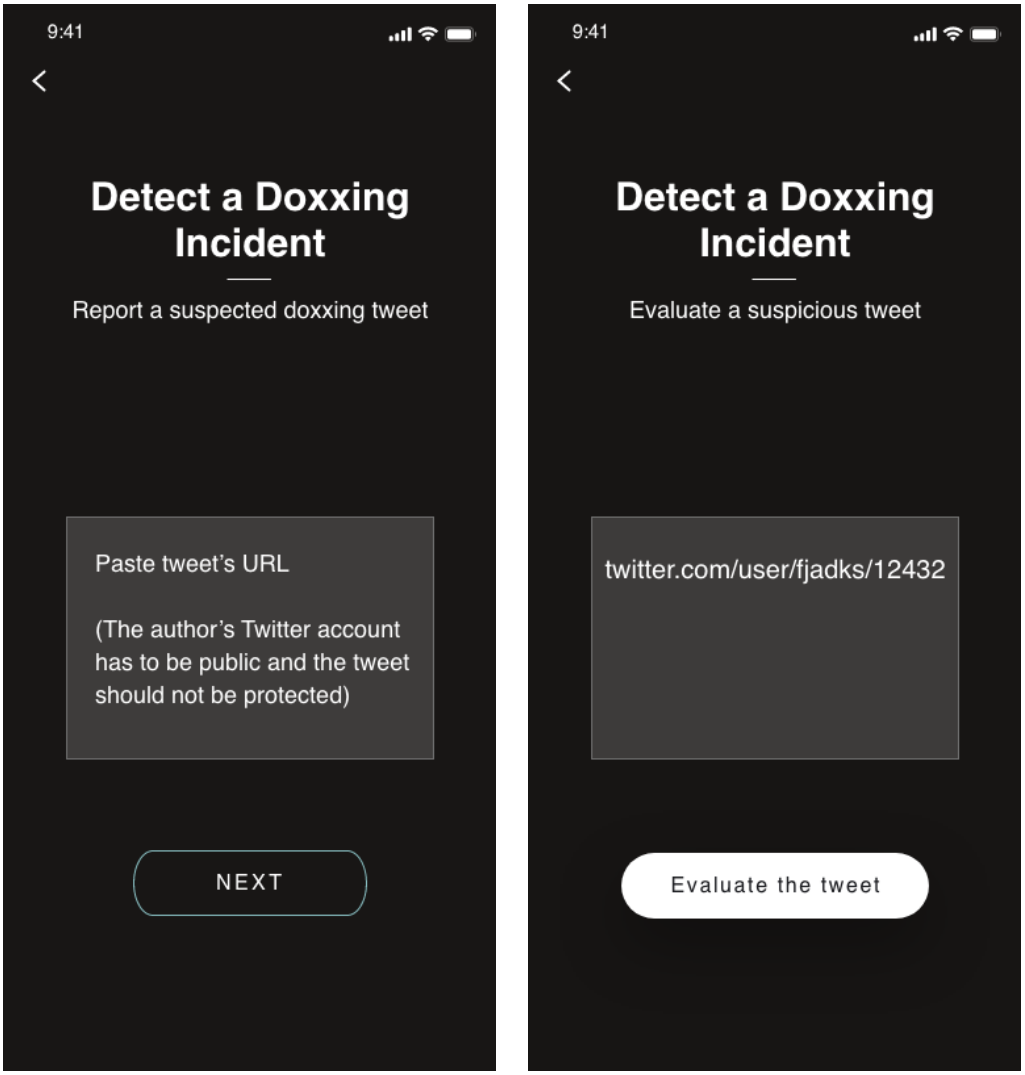


Figure 15: Prototype alt-boards



(a) (b)

Figure 16: Prototype alt-boards

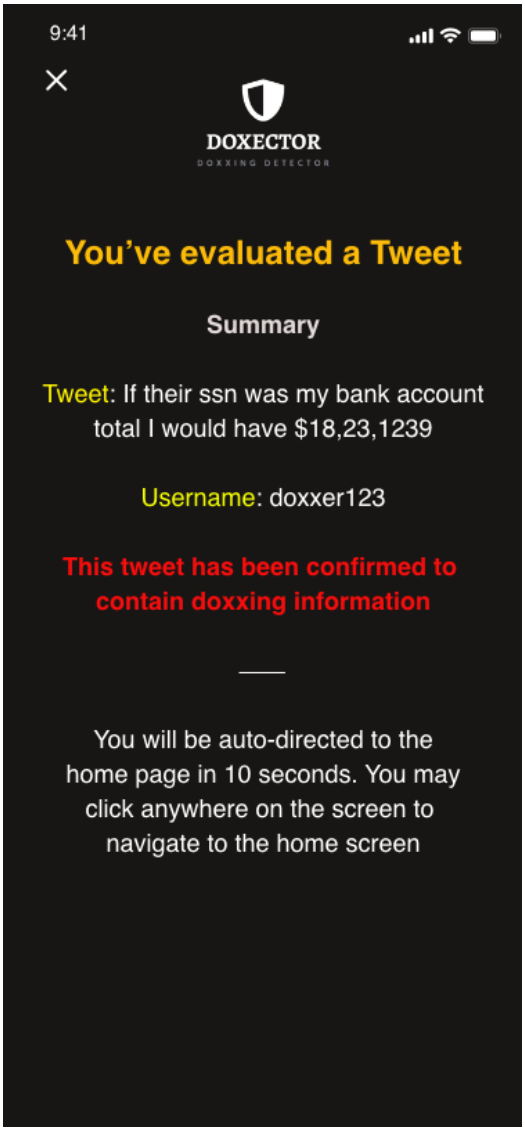
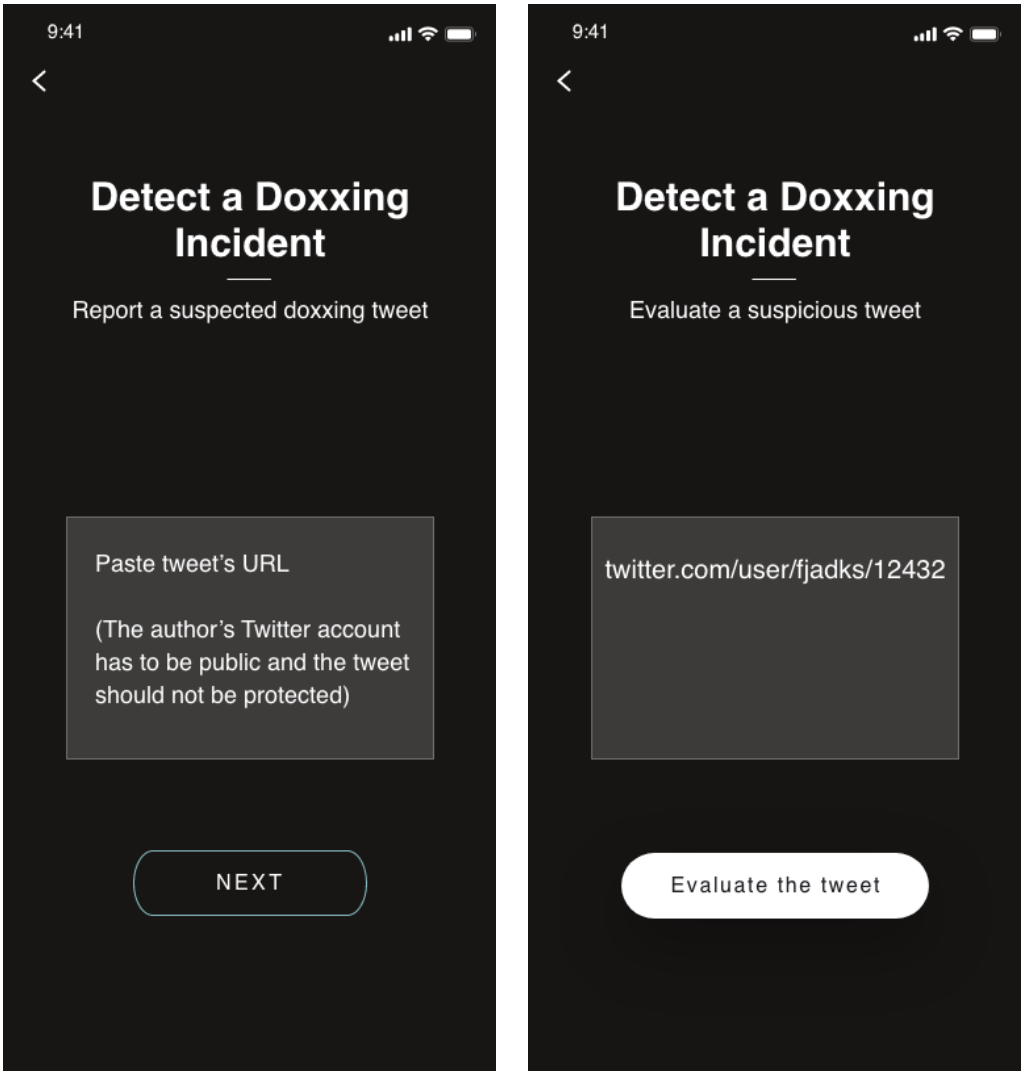


Figure 17: Prototype alt-boards



(a) (b)

Figure 18: Prototype alt-boards

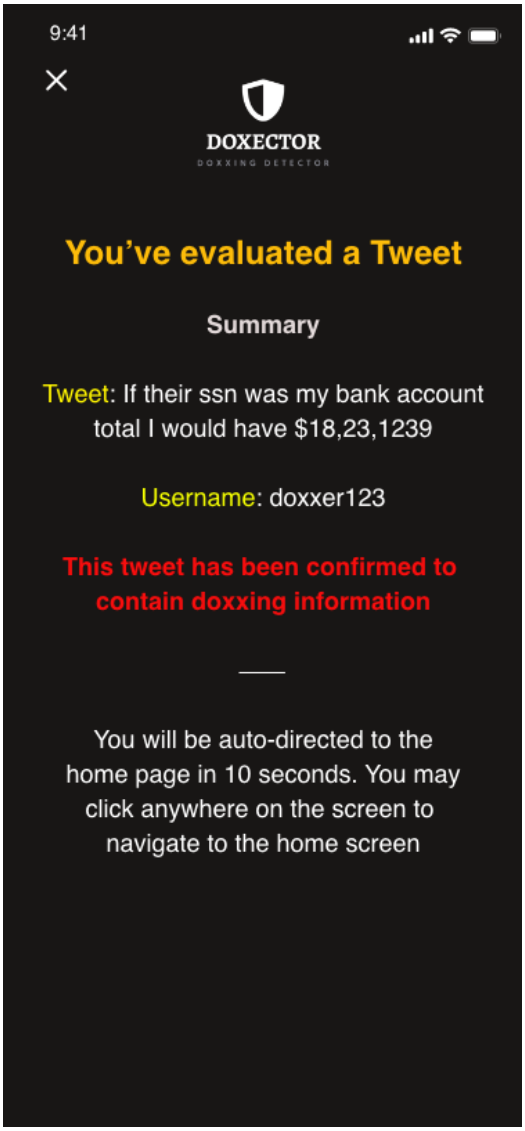


Figure 19: Prototype alt-boards