

Elaine Kirwan
R00138548
Let's Talk
Final Year Multimedia Project
2019

Abstract

Introduction

Let's Talk - an interactive e-learning application that deals with the topic of Sex and Puberty. This app is a contemporary tool for Parents to aid them in educating their Children about Puberty and Sex, Let's Talk will be an application targeted towards 8-12 yr olds and their Parents, taking the form of an interactive storybook, with characters guiding the child along a journey from childhood to adulthood. The idea behind the application is to make the learning of sexual education for children modern and safe. There is a lot of unsuitable content out there on the internet that is easily accessible for young children, Let's Talk aims to create a safe and relaxed learning environment for kids to learn about the changes that will occur throughout Puberty.

Inspiration for the project came from a TED talk by Alexis Jones called 'Locker Room Talk - Says who?', in which she discusses her work giving talks to young male athletes about respecting women. "We know that the majority of these young men learn about Sex through porn" (Jones, Alexis 2017) This inspired us to have a conversation around Sexual Education and our own personal experiences with the system currently in place. We all agreed that we had not had the best experience with Sex Ed in our own childhoods. After extensive research we observed that there was a lack of modern resources to help parents tackle the topics of Sex and Puberty at home. We proceeded to survey both our peers and a number of parents who have had a child of theirs go through puberty to get feedback on what they think of the current education system and their personal experiences. We surveyed 58 of our peers and of that 68% of them felt that there were grey areas in their Sexual Education. 44% of this group learnt most of their knowledge about Sex and Puberty from friends and 31% learnt most from the Internet and TV. We also surveyed 18 parents and 84% of those surveyed would be willing to have conversations with their child around the topics of puberty and sex. This feedback showed us that not only is there a need for better resources to address with this topic, there is a willingness amongst parents to provide their children with a resource like 'Let's Talk'

There is an increasing awareness nationally about the importance of Sexual Education and how the level and quality of Sexual Education received can affect a person's sexual behaviours and activities as they progress through life. Research shows a decreasing amount of parents having conversations about Sex and Puberty. "Parental involvement in sex education can serve to reinforce, and often supplement, key messages received about sex in school and

can impact on their child's sexual health in the future . Recent national research found that fewer parents surveyed in 2010 have talked to their children about sex and related topics (70%) than those surveyed in 2003 (82%)" (Kelleher C, Smith G, Bourke A, Boduszek D, McBride O, Morgan K.,2013) Let's Talk provides a tool that makes it easier for a parent to have vital conversations with their children about Sex and Puberty. Let's Talk takes a new and interactive approach to a curriculum that has been taught in schools for decades. The application will take the form of an interactive storybook featuring fun relatable characters for the child to choose themselves. The application covers the same topics surrounding Sex and Puberty that are included in the primary school curriculum for RSE. The Unique Selling Point of the application is how these topics are addressed. Characters guide users through a journey from childhood to adulthood, providing them with not only factual information, but also contextual scenarios and tips on how to handle puberty. Users will be tested on their knowledge of the topics they have covered through short quizzes and games, these will help reinforce the users knowledge and keep them engaged in the application.

The identified learner profile for the users of our application are children aged between 8 and 12 with basic literacy skills, who are comfortable using technology for both fun and learning. We aim to create a resource that allows the learner to interact with the information in a safe and relaxed environment, where they don't have to feel awkward or shy. Hence, a mobile application became our solution. The application will include animations, and informational text covering the subjects of sex, sex hormones, puberty, changes for girls, changes for boys, periods, pregnancy, birth and finishing with consent. The application will be built as a Progressive Web Application.

Individual Responsibilities

Over the course of this project my main responsibility was technical side of developing our application. My primary role within the project involved researching methods of developing an application in order to determine which would be the best approach for 'Let's Talk', taking into account the functionality required. This process began with researching different methods of application development and then experimenting with each of these techniques to establish their pros and cons. Initially, evaluation commenced with three methods, native iOS development, native Android development and Web Hybrid Development using Phonegap. After some further development and research, the term Progressive Web Application came to light. After further research into Progressive Web Applications, it was determined that this method was the most sensible solution for 'Let's Talk' going forward. After this had been established, my role was then to develop the application using both the knowledge gained from earlier studies, research and ongoing research. Using the knowledge of HTML, CSS, Javascript and PHP gained over the duration of my time in college, along with knowledge gained from personal research, I developed a working prototype for 'Let's Talk'. This involved coding the application framework and ensuring user can navigate through the application with ease. It was necessary to construct both an application shell model and a framework for the application itself. This involved creating

multiple pages for the application, some of which were reusable for several aspects of the application. Initial development began with the homescreen. This consisted of an animated logo, login and sign up options. A manifest JSON file was created to facilitate a full screen experience once the application was added to the homescreen. The application requires pages that support narration audio and interactive graphics, this was facilitated using HTML buttons that called a Javascript function providing audio and CSS animations when the user presses them. A page that tests the users learning of the materials was also required. This was facilitated using a jquery to pull questions and possible answers from an array and display them on the page in a random order. Other pages required include a diary page that logs a users progression through the application, and pages which support the animations as MP4's and allow to user to navigate to and from these animations. This role not only required development of these pages but also testing to ensure that the website functions as intended on the device.

My role within the project was determined by a number of factors, including prior experience with web development modules, personal interest in web development and ability within the area. I have always succeeded in web development modules and am the strongest member of the group in this area, so from the outset of the project in semester one, I took the lead on the practical development of the application. Although at times, web development can be challenging, college projects to date have equipped me with the skills necessary to solve any issues that may occur. There is a plentiful supply of online documentation when it comes to learning new web development techniques. I was confident from the start that with my prior knowledge of web development and my research skills I would be able to create an effective solution for a prototype of the 'Let's Talk' application.

Methodologies/Methods

'Let's Talk' has been developed as a progressive web application. Developed using web development languages, predominantly HTML, CSS and JavaScript, the application can be accessed via a URL, and once the site is added to the homescreen, the application resembles a native application. Distributed using a URL, the need for the application to be on the App Store is eliminated and the application can be downloaded onto any platform through the URL. Initially development of this application was intended to proceed using PhoneGap, an adobe application that allows for easy development and testing of web hybrid applications. After research into several methods of application development, a web hybrid application seemed like the most viable option for 'Let's Talk', see '[Research on Application Development](#)' - blog post that details this research. Consideration was given to native development for both android and iOS platforms, as well as developing the application as a web hybrid.

Demographically speaking iOS is a platform generally associated with more affluent users, it is the prominent platform in the US and it's users are more inclined to spend money on applications. (Teleman, Costi 2017) Android is the prominent platform in developing countries and amongst users who are spend money more consciously. Android is also much more prominent globally. (Katariya, Jayanti 2017). Initially it was intended that the 'Let's Talk' application would be developed primarily for the iOS platform, however there was a technical

barrier to this method, as the Xcode application used for iOS development is only available on Mac. There were solutions for this issue such as using the computers available on campus or renting a mac virtually at home, but as these options would be restrictive with regards to when I could work on the project could be completed, it was logically to move in another direction.

Android Development requires a program called Android Studio, which can be downloaded on any computer, windows or mac, and relies on the scripting language Java. The main issue expected with developing in Android Studio, is the complexity of Java as a language. Taking into consideration the timeframe of this project, learning Java with no prior knowledge of the language and understanding it to a sufficient level for it to be possible to develop a full application, would be out of the scope of the project.

Phonegap is an open source software produced by adobe that allows you to create mobile applications using HTML, CSS and Javascript. Initial interest in this platform for development arose for several reasons, the first being the fact that it would require and build upon the knowledge of web design already gained as a result of this course. This would also allow quick develop a prototype quickly. Additionally, using this tool the application could be built for both iOS and Android platforms concurrently. phonegap and had no issues with the tool. I hence concluded that developing the application using HTML, CSS and JavaScript through phonegap would be our best option. "PhoneGap is an application framework that enables you to build native applications using HTML and JavaScript. Think of PhoneGap as a webview container that is 100% width and 100% height, with a JavaScript programming interface that allows you to access operating system features." (Natili, Giorgio 2013). In essence PhoneGap allows you to use web development languages HTML, CSS and JavaScript, to develop and test mobile applications for both Android and iOS. PhoneGap is "an open source, cross-platform framework for developing mobile applications. PhoneGap allows developers to leverage web development skills— HTML, CSS, and JavaScript— to developed native applications for iOS, Android, BlackBerry, and many other platforms with a single codebase." (Lunny, Andrew 2011) There are three main types of applications available, native apps, web apps and hybrid apps. There are advantages and disadvantages to each method of application development. Native apps are made specifically for the OS they are intended for. Native applications can store more data offline and have more access to the devices hardware and OS features. These apps are more costly because they have to be coded in the language corresponding to each specific platform.

Web Apps can be developed using one common language for all platforms, however they have limited functionality and access to the devices OS. "Web apps are built using web standards, so it's possible to have a common code base for all the supported devices and it's much easier to find developers with the appropriate skills. On the con side, a web app is not able to access the entire device's hardware and OS features and is not installed on the user device, which means that users cannot access it when the device is not online."(Natili, Giorgio 2013).Hybrid applications are a mixture of web apps and native applications. "A hybrid app can be viewed like a web app on steroids; in fact it uses web standards and can access most of the device's hardware and OS features"(Natili, Giorgio 2013). After attempting to use Phonegap Developer app on several device, including a new tablet, it was discovered that there was a long list of requirements for the application to run properly. After bringing this issue to a web

development lecturer it was suggested development of the applications continue as a progressive web application, using the same principles and techniques as with Phonegap, but distributing the application via a URL.

People gravitate towards native apps because of their speed, reliability and immersive experience. However in the last few years web technology has advanced to the level where it is possible to achieve a similar speed, reliability and experience as native technology using web standards. 'Progressive App design offers us a way to build better experiences across devices and contexts within a single codebase' (Russel 2015). The term Progressive Web Applications represents websites that can act and react like native apps, providing offline service, fast interactions and immersive full screen experiences. Progressive web applications are essentially websites that can be added to the homescreen and will then give a user experience that closely resembles that of a native application. "While a PWA can be treated like a standard application on a device, the ability for it to be run from a uniform resource locator (URL) makes it easy to use the application on any device with a browser."(Frankston, Bob 2018). Progressive Web Applications are different from standard websites partially because they offer users an offline experience. When a website fails to load, or the connection is poor, the website will show a connection error, whereas an application will generally load it's interface and some content, even if that content isn't new. To create a similar experience to native applications, progressive web applications need to emulate the offline experience native applications provide. This is achieved using the app shell model and service workers. The App Shell model consists of the HTML, CSS and Javascript required to provide the user with the basic application interface. "An application shell is the secret to reliably good performance. Think of your app's shell like the bundle of code you'd publish to an app store if you were building a native app. It's the load needed to get off the ground, but might not be the whole story." (Gaunt, M. Osmay, A. 2019) Service workers are Progressive Web Applications solution to offline functionality, they cache previous loads of the application shell and use this when the network isn't available. Essentially, once a user loads the application once the application shell is cached, then on subsequent visits to the application, the service worker retrieves the application shell from the cache, stored locally on the device. This not only allows for the application shell to be accessed when the device is offline, it allows for faster loading times. This can be referred to as developing for offline-first. "With offline first we assume of line, and do as much as we can with local content, the try and get content from the network, and the more we get to render without a connection, the better. So you should think of the network as a piece of progressive enhancement, an enhancement that may not be there" (Archibald, J. 2016)

Discussion

Appendices

References

Kelleher C, Smith G, Bourke A, Boduszek D, McBride O, Morgan K. Sex in Ireland in the last decade: sexual health research and its policy implications. IRCHSS Policy Brief. Dublin: Royal College of Surgeons in Ireland, 2013. [online] Available from URL : <https://epubs.rcsi.ie/cgi/viewcontent.cgi?article=1053&context=psycholrep> (accessed 7/11/2018 and again on 25/11/2018)

Jones, Alexis “Locker Room Talk, Says Who ?”, TEDX University of Nevada [online] available from URL <https://www.youtube.com/watch?v=ICA6EF3y23k> (accessed 31/08/2018 and again 25/02/2019)

Jayanti Katariya (2017) Apple Vs Android – A comparative study 2017 [online] Available from URL : <https://www.moontechnolabs.com/apple-vs-android-comparative-study-2017/> [accessed 02/02/2019 13:50]

Teleman, Costi (2017), Android vs iOS App Development: A Comparison and Decision Guide [online], Available from URL : <https://dzone.com/articles/android-vs-ios-app-development-a-comparison-and-de> [accessed 02/02/2019, 14:45]

Natili, G 2013, PhoneGap Beginner’s Guide, Packt Publishing Ltd, Birmingham. Available from: ProQuest Ebook Central. [1 March 2019].

Lunny, Andrew. PhoneGap Beginner’s Guide, Packt Publishing Ltd, 2011. Available from: ProQuest Ebook Central. [1 March 2019].