ISSN: 2348 - 2117

International Journal of Engineering Technology and Computer Research (IJETCR)
Available Online at www.ijetcr.org
Volume 4; Issue 6; November-December; 2016; Page No. 65-69

iOS MOBILE APPLICATION DEVELOPMENT WITH FULL CASE STUDY: WEB-ARCHIVE

Prakash Kawar Aswat[1], Girdhar Gopal Singh[2]

[1] Assistant Professor, Jodhpur Institute of Engineering &Technology, Rajasthan, India
[2] Jodhpur Institute of Engineering &Technology, Rajasthan, India

Abstract
This paper is on an iOS mobile application which is developed using swift programming language. This app is to simplify online browsing and content reading through its best user experience. .For example if you read an article online and want to save it offline for future purposes without online ads and don't want to keep tabs always open. This seems very easy but it becomes very annoying when you want to read some big article,book and so on, as they have many parts and you want them to be stored offline and read whenever you want with or without the internet.This app also allows user to read latest online available news and articles from trusted sources.

Keywords: Offline Data download, More Reading, Web Archive, News.

I. INTRODUCTION

This is an iOS application that enables its user to store web pages on mobile offline to read without the internet.. It is very convenient as its name suggests,it includes many different applications and challenges. In this project the main goal of the application is to develop an iOS app, Which stores Websites in local storage via TheWebArchive[1],Userdefaults[2] and Core Data[3] by opening the URL using Webkit[4]. Currently we have some alternatives to this application but most don't work properly or have paid subscriptions. We need to make this project free of cost by only using Open Source[5] packages and then this case study is done for this project to make this application for a large group of people and make their life easier in the absence of the internet. The WEB ARCHIVE iOS application is developed with the best user interface by implementing basic and simple parts for now and then later adding more features into this application without filling too much data without legal(Copyright)[6] planning. The application is very useful when there is no internet connection or we want to store articles to read later.. The downloaded pages will be available to view even when the original site is not available on the internet or the application is uninstalled as the data is being saved in File Storage[7]. We will use the Waterfall Model of Development for the application to design a working app with all needed features and then we will add more and more features later by keeping feedback in mind

II. TOOLS REVIEW

A. XCode By Apple

Apple-Xcode[8] is an integrated development environment (IDE) for macOS(operation system of apple computers), used in application development of apple products like tvOS,iOS, macOS, iPadOS, and watchOS. It was first introduced in 2003,and it keeps growing with launching new versions, the latest stable release is version 13.2.1(13C100), released on December 17, 2021, and now it is available through Apple's XCode website and macOS app store. We are using XCode Version 12.3 for the project. The Apple XCode 13(Latest version) includes various features like XCode Cloud that can be helpful in collaboration with other developers on platforms like Github and Bitbucket.

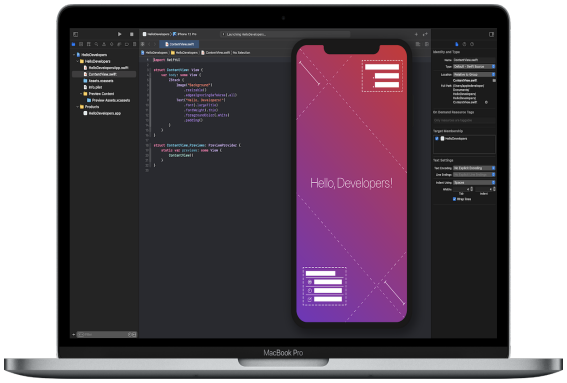


Fig. 1: XCode by Apple

Corresponding author: Prakash Kawar Aswat

B. Frameworks: UIKit,WebKit.

While developing an app in iOS or MacOS whenever an app like a browser is very important and it has to be developed. WebKit is a framework which works with Web Engine.

C. UIKit

When we have to make cool web user interfaces in iOS we can use the UIKit[9] framework. It is very fast.

D. WebKit

WebKit is developed by Apple.It is an Open-Source[5] Web Engine and this is used by many popular applications on macOS, iOS, and Linux.

E. Dependency Manager: Cocoapad, Swift PackageManager

What is a Dependency manager?

Dependency manager is used in the automated fashion of resolving packages and their internal libraries used in the development process.Like if we traditionally try to install a package "X" which imports classes of package "Y"; We will not be able to use it as only single or standalone package "X" will be installed. There the Dependency Manager is useful for automatically installing additional packages. There are Cocoapod[11], and Swift Package Manager[10] for managing packages in iOS app development when using Swift[12] language.

A. Programming Language: Swift 5.5

Swift is a programming language developed by Apple Inc.

and this has some very useful features like general-purpose, multi-paradigm, compiled programming.It is also available as Open Sourc.In Worldwide Developers Conference (WWDC) it is First released the date was 2 June,2014. Swift was developed as an alternative to Objective-C, as Objective-C hasn't been used in much progress since the 1980s and not able to provide the latest technology features required to create modern iOS, and macOS applications. The latest stable version of swift is 5.4.1 released on 25 May, 2015.It has powerful pattern matching with a modern, lightweight syntax, and type inference allowing complex ideas to be visualized in better way.So the code of swift language is simple,fast and easy to read. For this project we will be using Swift 5.3 version

F. User Interface: SwiftUI

When We need to Design responsive, and cool apps SwiftUI[13] can be used across all platforms provided by Apple. By using combined Power of Swift— SwiftUI we can make the code responsive so it enables different views according to different platforms. It also offers many tools and APIs to get best user experiences for all users, on any Apple products.

G. HTML Parser: Fuzi

With XPath & CSS support, Fuzi[15] is a fast & lightweight XML & HTML parser. Fuzi is based on Swift port of Mattt Thompson's Ono, With moderate class & interface redesign following standard Swift conventions it mostly uses low level implementations, along with several bug fixes.

III. METHODOLOGY

We used the Waterfall Methodology for this project. Iteration Methodology mainly used to first make MVP(Minimum Value Product) and then use the provided feedback by it to make application's good version.

DFD and CFD

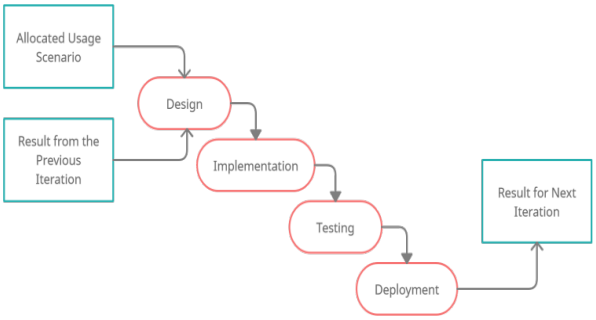


Fig. 2: Iteration Workflow

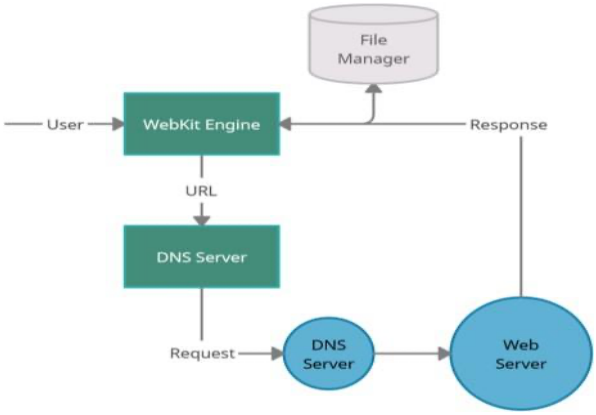


Fig. 3: Data Flow Diagram

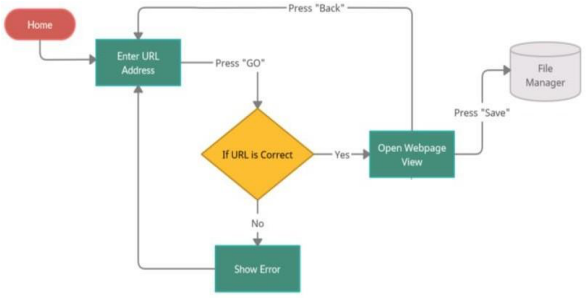


Fig. 4: Saving webpages -Control Flow Diagram

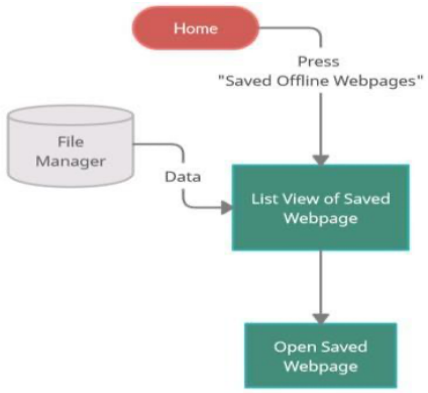


Fig. 5: Viewing web pages - Control Flow Diagram

A. App Component

1. Home Pages

Paste the URL on the input tab and press “GO”.

1. WebKit Web Viewer

To view the webpage we are using the WebKit Engine. There is also an option to either revert back to the homepage and enter a different url or save the current webpage

Fig. 7: Web View

1. Save

On the Web View you can Click on Save Button to save the Web Page. Saving the current page will download the page locally and add it to the Offline Saved Website page which the user can navigate from the home page.

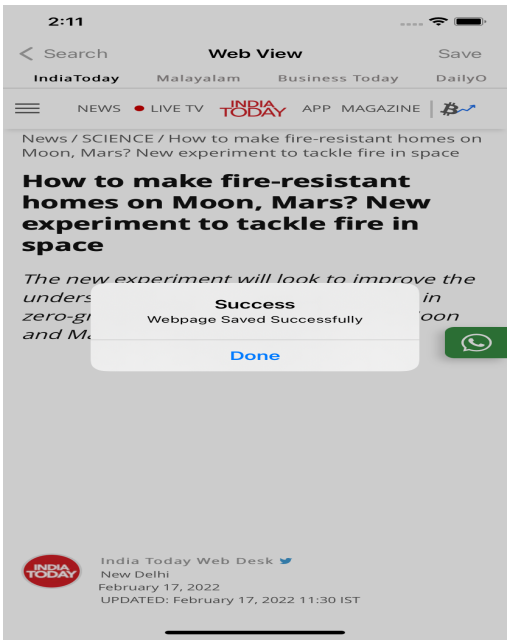
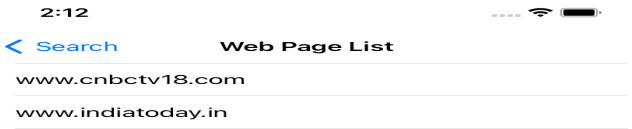


Fig. 8: Successfully Saved

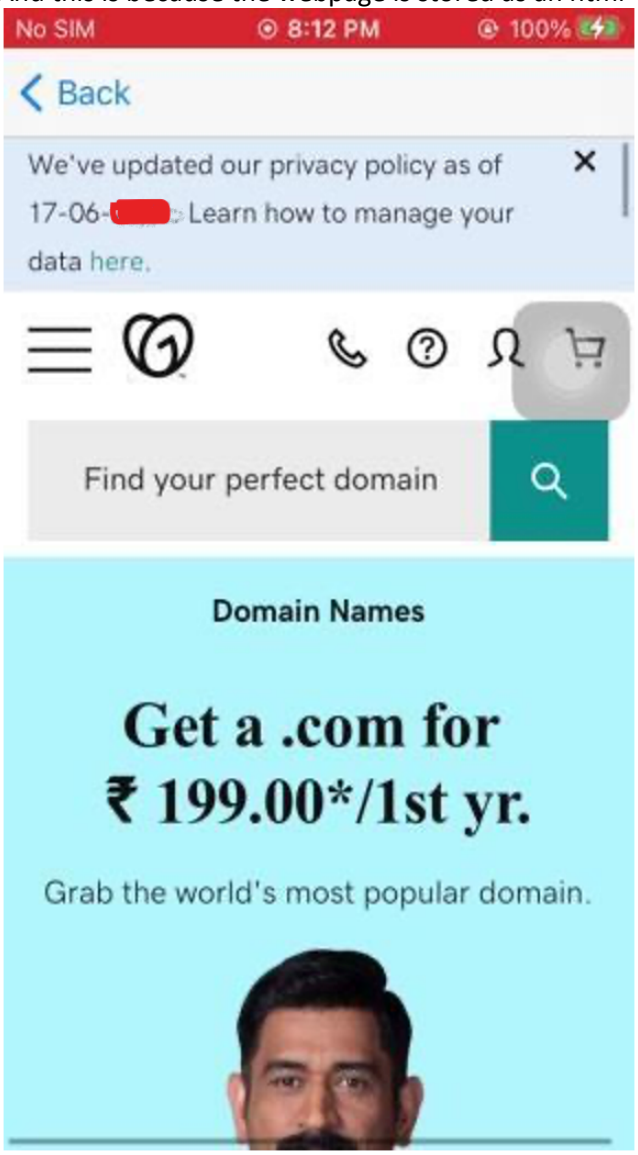
1. Offline Saved WebSite

Firstly Disconnect the mobile from the internet. Then click on Offline Saved Website Button from the Home Page. In the Offline Saved Website you will be able to see the list view of all the saved websites and articles



5. Saved Pages In the list you can now open any of the websites and read it whenever you want. The

webpage saved persists even when you kill the app. And this is because the webpage is stored as an html



document with all attached files in the File Manager with the in-built feature provided by Swift.

Fig. 9: Offline Saved Website ListFig. 10: Saved Web Page

IV. CASE STUDY AND SURVEY

A. Problem Statement

Most Often we find ourselves in a situation where the internet is not stable or we find an article but don't have enough time to view the article and want to save them offline. so, we can read it in the airplane or at our discretion. Some people also like to multitask and do their chores while listening to music or in this case listening to the article. But is this feature popular and how many users may want this feature. The problem with Text-to-Speech is that it doesn't offer much control.

A. B. Drawback: As case study pointed out, the default Text-to-Speech feature only works per document and we want it to work like a playlist with features such as queue, speed of speech*Solution*

To resolve the above problem we will be using a Podcast API that sends text to a Server then takes mp3 format audio of that text and thus controlling the mp3 format and implementing the above task becomes easy *Survey Conclusion*

We found out that this application is very niche based and not all people found it useful as they mainly good portion of people don't read articles; which is not at all bad news as we also found out that people who work in professional industry often do require to read lot of articles and may want the possible solution that our application may provide.

People also added to what other features can be added like:

1. Rating of an Article.
2. Suggestion Feed of Related Article.
3. To be able to share users saved lists to other users.

Subject of Interest for Recommended Feed.

V. DISCUSSION AND FUTURE WORK

A. Better User Interface and User Experience

Everyone knows how important it is to provide great User Experience, for the user to keep using the application. From the user's perspective only two things matter about the application: how it looks and how it feels. Users don't care how great of a technology or what type of complex structure is used to give the final product. They only care about look and feel. That is the reason why one of the main goals in the future is to make the application provide a great user interface(UI) and user experience(UX).

B. Additional Features

Future Goals:

1. Adding Translation Feature for the saved article when online.
2. Able to create a podcast out of articles to be able to listen using some human-like voices rather than default robotic voices.
3. Scroll on a saved article to be able to follow where the text-to-speech is at.
4. Give recommendations or related articles with the saved pages when online.
5. Checking Website Certification to determine f web pages can be saved offline or not.

Creating an Android, and Web Application and being able to sync data between them.

Future Challenges There are three challenges our team foresight in completion of project: 1. Translation Feature to be implemented as all the

good translation APIs are paid and free Translation API is not good enough. 2. As not being an expert, we can't really tell if we are breaking any law when saving online web pages offline as it can generate copyright infringement. We have done some research and found out that it's actually ok to save websites available on public domain. So, we have to make security changes so users aren't able to store websites that have strict rules or are out of public domain. We can achieve this by checking the Website Certificates but as of now we don't know how to do it. 3. Having a Human-like Voice for Text-to-Speech.

VI. CONCLUSION

We have created the iOS application which provides the user to save a website or article for offline storage and be able to read them at discretion or at an uncomfortable time when the internet is not available. We are very happy that we are able to contribute to the idea of how we consume online content and may possibly allow people to be given alternative options for how they choose to read an article. Currently the application provides basic features and has much more to improve where in the field of User Interface and User Experience and possibly implement additional features.

ACKNOWLEDGMENT

It gives us great pleasure in writing this paper. Firstly, We would like to show our sincere gratitude toward our guide "Mr. krishna kantt Lavania" and all the supporting personnel of the Department of Computer Science & Engineering, (JIET, Jodhpur) who provided us with their experience and expertise. Also, a great and well-deserving thanks to Apple and Open Source Community for creating and maintaining these high-end technologies.

REFERENCES

1. https://developer.apple.com/documentation/webkit/webarchive
2. https://developer.apple.com/documentation/foundation/userdefaults
3. https://developer.apple.com/documentation/coredata
4. https://developer.apple.com/documentation/webkit
5. https://www.oreilly.com/radar/topics/open-source/
6. https://copyright.gov.in/
7. https://developer.apple.com/documentation/foundation/filemanager
8. https://apps.apple.com/us/app/xcode/id497799835?mt=12
9. https://getuikit.com/docs/introduction
10. https://swift.org/package-manager/
11. https://cocoapods.org/
12. https://swift.org/blog/swift-5-3-released/
13. https://developer.apple.com/xcode/swiftui/
14. https://devopedia.org/dependency-manager
15. http://cezheng.github.io/Fuzi/