

Armoneam Tesfaye

# Using PhoneGap to Create a Wish List Application

Helsinki Metropolia University of Applied Sciences

Bachelor of Engineering

Information Technology

Thesis

21 April 2017

Author(s) Title	Armoneam Tesfaye Using PhoneGap to create a wish list application
Number of Pages Date	37 pages + 3 appendices 26 May 2017
Degree	Bachelor of Engineering
Degree Programme	Information Technology
Specialisation option	Software Engineering
Instructor(s)	Jaana Holvikivi, Principal Lecturer
<p>The aim of this thesis project was to explore different mobile application development approaches and give more emphasis on a hybrid mobile application development process. The hybrid application development approach was compared to the native application development and web application development approaches and the advantages and disadvantages of each approach were explored.</p> <p>The development framework that was used was Apache PhoneGap, which is an open-source mobile development framework. The user interface was built using HTML, CSS and JavaScript programming languages.</p> <p>The outcome of this project was a hybrid application which is capable of storing user data on the local storage of a device. The application uses features such as camera and social share and it can be installed on multiple platforms.</p> <p>From this project it can be concluded that hybrid application development has remarkably changed the field of mobile application development. The benefits and drawbacks of a cross-platform application over their rivals were discussed.</p> <p>The application that was developed for this project can be further improved by the use of a remote server to host user data and implementation of unique user profiles for users, so that it will be possible to access stored information from another device.</p>	
Keywords	hybrid application, Android, IOS, Apache Cordova, PhoneGap, mobile platforms

## Contents

1	Introduction	1
2	Mobile Phone Applications Development	2
3	Mobile Phone Platforms	3
3.1	Android OS	3
3.2	iOS	4
3.3	Windows Mobile OS	4
3.4	Blackberry OS	5
3.5	Firefox OS	5
3.6	Sailfish OS	5
3.7	Ubuntu touch OS	6
4	Mobile Applications	6
4.1	Web Applications	7
4.2	Native Applications	8
4.3	Hybrid Applications	9
4.4	Comparison Between the Application Types	10
5	Technology Implemented	11
5.1	HTML5	11
5.2	CSS3	12
5.3	Apache Cordova / PhoneGap	13
5.4	JavaScript Language and Libraries	16
5.4.1	jQuery Library	17
5.5	Ripple Emulator	18
6	Implementation	21
6.1	Description	23
6.2	Main Components	24
6.2.1	Add a Wish	24
6.2.2	Wish List Update	26
6.2.3	Take a Picture	27
6.2.4	Social Sharing	30
7	Testing	31

7.1	Testing Using Emulator	31
7.2	Testing Using a Real Phone	32
7.2.1	iOS Device	33
7.2.2	Windows Mobile Device	33
7.2.3	Android Device	34
8	Result and Discussion	35
8.1	Positive Outcomes	35
8.2	Challenges	35
8.3	Future Improvements	36
9	Conclusion	37
	References	38
	Appendices	
	Appendix 1. Code Listing from config.xml	
	Appendix 2. Code Listing from index.html	
	Appendix 3. Code Listing from camera plugin camera.js	

## 1 Introduction

The time is for mobile devices. Mobile devices are the new normal in our life affecting every aspect of it. There are different mobile devices ranging from mobile phones and tablets to wearable devices. The popularity of these devices is growing, and the number of users around the world has increased dramatically in the last decade. All these mobile devices have something in common: they all have applications running on them so that they will be easy and efficient to use.

Among mobile devices the most popular one is a smart mobile phone. Smart mobile phones can be used to make calls, send text, take pictures, shop and virtually most of the things that are possible to do on the internet. Even though all smart phones have a similar purpose, they have different operating systems. The different operating systems have different ways of working and are built using different programming languages making the applications installed on them to have different source codes.

The fact that different operating systems demand different source code has created a challenge for application developers to develop applications separately for each operating system. This process demands a longer time for application development, additional costs for companies and an upkeeping challenge for developers. Many companies want their applications to get to as many users as possible, which means a need for their applications to be available on as many platforms as possible. This causes additional costs for the development and the maintenance of the applications.

Thus, the challenge has created the solution of developing applications which can run on different platforms. These applications are called hybrid applications. Even though hybrid applications have the advantage of running on many platforms, they also have their drawbacks compared to native applications. In this thesis, I have explored different platforms, different application types and finally the development process of a hybrid application.

## 2 Mobile Phone Applications Development

Mobile applications development is the process of developing applications software which is specifically designed for mobile operating systems and intended for handheld devices such as mobile phones or tablet computers [1]. Mobile users use mobile applications to access many different services online for their day-to-day activities. Mobile applications are preferred to be used instead of access through mobile browsers, because of their ease of use and more user-friendly user interfaces.

The number of mobile phone applications has risen meteorically in the last few years thanks to their popularity and the rise of the number of mobile phone users. The growth of the number of mobile users has been significant while the number of desktop users has been in decline. Mobile technology trends report from KPCB (Kleiner Perkins Caufield & Byers) that the time users spent on mobile phones is outstripping the desktop time by a 9% margin. That is the time spent on mobiles is 51% compared to the 41% of the time that is spent on desktop computers. [2.]

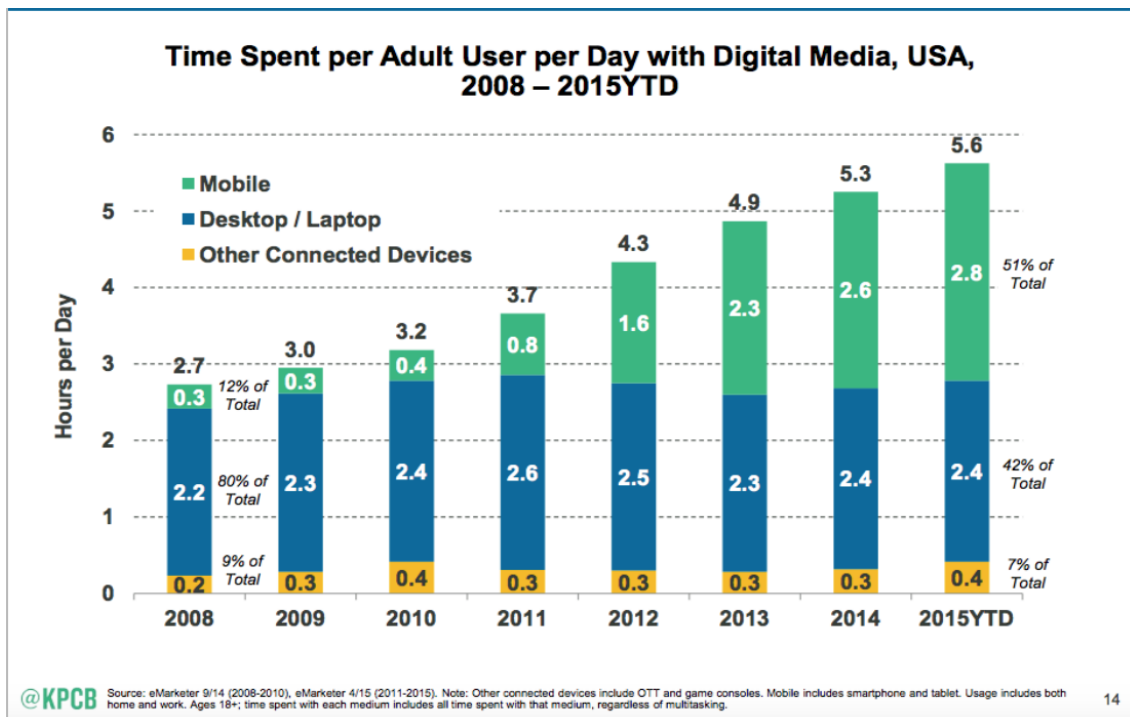


Figure 1. Time spent on devices compared. Reprinted from Meeker (2016) [2]

As Figure 1 shows the amount of time spent on mobile devices by mobile users has grown significantly in the USA through the past few years compared to the time spent on using desktops or laptops.

### 3 Mobile Phone Platforms

A mobile phone platform also known as a mobile phone operating system is a collection of data and software instructions on mobile phone hardware. It manages the hardware and enhances the ability of mobile applications software on the hardware. It controls and manages the different functions of the device such as multimedia, internet connectivity, Bluetooth connectivity and several phone accessory connectivity.

Table 1. Different mobile phone platforms. Reprinted from IBM (2012) [f13,6]

	Apple iOS	Android	Blackberry OS	Windows Phone
Languages	Objective-C, C, C++	Java (some C, C++)	Java	C#, VB.NET and more
Tools	Xcode	Android SDK	BB Java Eclipse Plug-in	Visual Studio, Windows Phone development tools
Packaging format	.app	.apk	.cod	.xap
App stores	Apple App Store	Google Play	Blackberry App World	Windows Phone Marketplace

Table 1 shows the different mobile platforms and the corresponding programming languages associated with them. It shows also tools for development as well as the App stores where the applications are available.

#### 3.1 Android OS

The Android operating system was invented and developed by Android Inc. and was later acquired by Google Inc. in 2005. Google in turn came together with other companies to form the Open Handset Alliance (OHA), which is now responsible for the ongoing development and growing of the Android operating system. [3.]

The Android OS is based on a Linux kernel and its code is written in Java for user interface, C programming for core programmes and C++ programming language. The software code is under the open-source software license. According to Android Inc. Android OS powers hundreds of millions of handheld devices in more than 190 countries. It is the largest installed operating system and the fastest growing platform [4].

### 3.2 iOS

Apple Incorporation's iOS is a mobile operating system which is designed and developed by Apple Inc. for its own hardware mobile devices. It is currently deployed on devices such as iPhones, iPads and iPods. It is the second most popular platform next to the Android platform.

According to Apple Inc. 27th WWDC in 2016 there are more than two million applications in the iOS App store and those apps has been downloaded more than one hundred thirty billion times since the app store was launched more than ten years ago in 2007. The iOS has up to seventeen million registered developers for its platform making it the second popular platform for developers. The software is a proprietary software which is written in C, C++, Objective-C and Swift. It has gone through different versions and the latest one is the iOS 10 [5].

### 3.3 Windows Mobile OS

The Windows mobile OS is a platform which is developed by Microsoft based on the Windows CE kernel, and is designed and implemented in a way that the mobile device will operate and feel like a Windows desktop operating system [6,229-231].

The first Windows mobile operating system started in the year 2000 for the device called pocket PC. It was named "Windows mobile" for the first time in 2003. Windows Phone, the successor of the Windows mobile, which was designed to target primarily the consumer market unlike its predecessor which was targeting the business market.

Windows Phone OS was superseded by Windows 10 Mobile in 2015. The Windows 10 mobile tends to give more attention to the unification of the Microsoft's pc and mobile user experiences and platforms. The software is written in C and C++ and its applications run on the mobile .NET version.



### 3.4 Blackberry OS

The Blackberry operating system for mobile phones is developed by Blackberry Limited or also known as Research in Motion Limited (RIM) for its mobile hardware devices. The operating system is written in C, C++ and Qt and its kernel is based on QNX Neutrino RTOS kernel.

The software is proprietary software for Blackberry smartphones. The latest version is called the Blackberry 10 and was released in 2016 [6,253-256].

### 3.5 Firefox OS

The Firefox mobile operating system is an open-source licenced operating system which was developed by Mozilla. It is based on the Linux kernel and the rendering power of the Firefox web browser. It is known for using open standards and highly utilises the standards of JavaScript and HTML5. The software is written in HTML5, CSS, JavaScript and C++.

The Firefox mobile was first officially released in the year 2013 but in December 2015 Mozilla announced that it will stop the development of smart phones [7]. Moreover, in September 2016 Mozilla announced the end of the mobile OS software development [8].

### 3.6 Sailfish OS

Sailfish OS is a Linux-based operating system developed by Jolla, the Sailfish and Mer-project communities and various open community members. The user interface was developed by using QML (Qt Modelling Language). [9.]

Sailfish operating system has also the ability to run Android Applications making it more flexible and easy to use. Since the Sailfish's libraries are based on the Android libraries, making it almost identical to run Android applications on Sailfish compared to running Android native applications on Android operating system that they were designed for [9].

### 3.7 Ubuntu touch OS

The Ubuntu touch mobile operating system is the mobile version of the Ubuntu operating system. It was designed to target smart phones which are capable or have the feature of a touch screen. It uses the Qt5-based touch user interface [10].

The Ubuntu touch is developed by Canonical Ltd. and uses Linux kernel making it easy to be implemented on many Android devices. Being open-source software and high flexibility in the programming language makes the Ubuntu touch an attractive choice to many developers and users [10].

The above mentioned platforms are not the only ones which have been developed through the time of mobile OS platform developments. It is good to note that how they are different and vary in many aspects, which makes it hard to choose one platform to develop an application. It will be time and money consuming to develop an application which can run on all platforms. In the future, I believe that the technology will come together to allow an application to run on all platforms without the need to worry about platform-specific issues.

## 4 Mobile Applications

Mobile Applications also known as Mobile Apps are application software programmes which are either pre-installed or later installed programmes which run on mobile phones so that users will be able to use their handheld devices to the fullest potential or to different purposes. Most mobile apps have a desktop counterpart. Even though most mobile Apps have the same purpose as their desktop version, it does not mean that they have the same way of working. In mobile phone devices the screen area is much smaller than the traditional desktop screen, making it more difficult for users to precisely touch the target that they want if the links or the buttons are very small and very close to each other. [11,41-44.] The simplicity that comes with the limitation of the screen makes mobile Apps easier and simpler to use, making them more user-friendly in this respect.

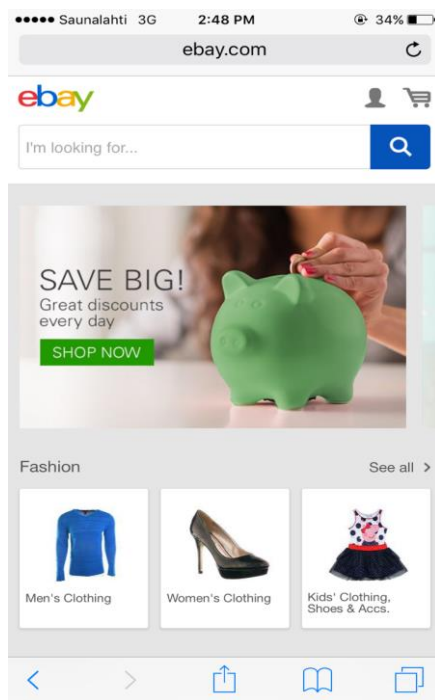


. Figure 2. Android and iOS Mobile devices native application examples.

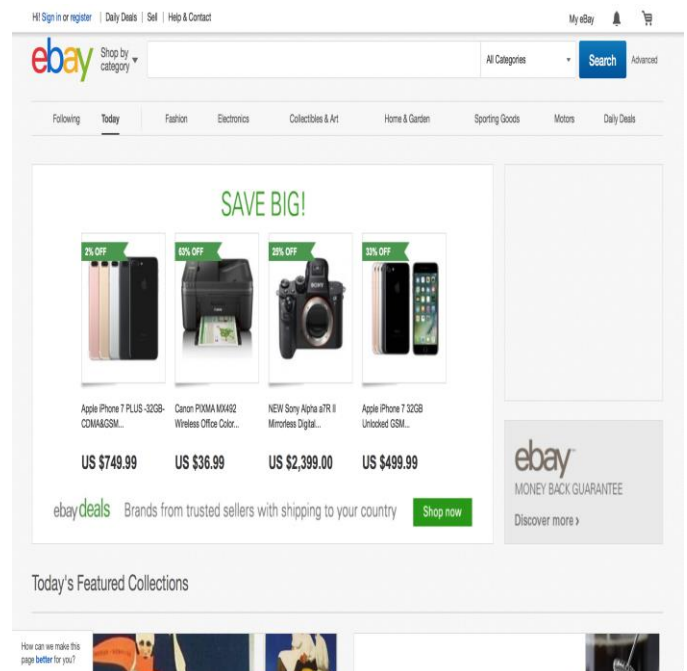
Figure 2 shows the pre-installed native Applications for an Android and an iPhone mobile phone platforms.

#### 4.1 Web Applications

Mobile web applications are applications software which are developed using web technologies. Even though they are used on mobile devices they are not hosted or installed on the mobile device rather they are hosted on a remote server and served to the user using standard protocols such as HTTP [12]. Mobile Apps typically run on web browsers and might differ how they look from one browser to the other because they are platform dependent. Those platforms support different versions of the HTML5. [11,41-44.] The most important feature that these web applications give to the user is that they tend to give the feeling and look of being the same with native apps, making web applications easier to use and more user friendly than their desk top counter parts.



A



B

Figure 3. Web application versus desktop browser version

As can be seen from Figure 3 A, the web app version of the online retailer company eBay is much simpler compared to the desk top version of the website. But on the negative side, the web app version might be difficult to navigate since all the navigation links are not upfront like the desktop version in Figure 3 B. Since the code that is used to write web apps adapts a standardized programming language, it makes it easier for fast development, easy maintenance and overall low cost for developers and owners [13].

## 4.2 Native Applications

Native applications are applications that are specifically programmed to run on a specific mobile device platform. They have binary executable files that are downloaded to the device to be stored locally [13,2]. Native applications are usually downloaded from an online application store such as Apple's App store, Android's Marketplace or by other means such as directly from online, for example from the application developer's site. Typically, native applications have the advantage of access of the APIs (Application Programming Interface) that are available from the operating system [13,3].

Since native application platforms differ from one another there is a negative side to developing a native application. The code which is written to develop a native application for one platform cannot be used to develop the same application for another platform, thus making the developing and the maintenance of a native application an expensive and lengthy process.

### 4.3 Hybrid Applications

Hybrid Applications are applications that are built with the blend of web technologies such as JavaScript, CSS, and HTML5. Their main difference from web applications which use the same technology is that they are nested inside the native application container that will allow them to access the whole set features that the device offers to native applications [13,6].



Figure 4. Application configuration. Reprinted from IBM (2012) [13,7]

Figure 4 shows different configuration styles of applications that will help them utilize the device features. Application developers have the choice to either develop their own channel between the browser and the device's APIs or use some ready-made solutions that are on the market. For example, PhoneGap is an open source JavaScript library interface that can be used in the development of hybrid applications so that they can access native device features that are not available for simple web applications. For this thesis project, I have used the PhoneGap interface to develop my own hybrid application.

#### 4.4 Comparison Between the Application Types

In a comparison between these three approaches, one can see that every approach has advantages and disadvantages or limitations. Choosing the right option might depend on the project type, financial limitations, time limitations and many other technical requirements. This section will try to show the advantages and disadvantages of each approach.

Table 2. Comparison of application types. Reprinted from IBM (2012) [13,8]

Feature	Native app	Hybrid app	Web app
Development language	Native only	Native and web or web only	Web only
Code portability and optimization	None	High	High
Access device-specific features	High	Medium	Low
Leverage existing knowledge	Low	High	High
Advanced graphics	High	Medium	Medium
Upgrade flexibility	Low (Always by way of app stores)	Medium (Usually by way of app stores)	High
Installation experience	High (From app store)	High (From app store)	Medium (By way of mobile browser)

In the evaluation of Table 2, native application development is preferred when there is already the knowledge of the native application development language, the target of only single operating system, need of particular native functionality and requirement of high user interface like games [13,8]. Web application development on the other hand is preferred when requirements such as development of a pilot app is needed, visibility in search engines is important and when the need to avoid third party application stores and instead directly distribute the applications for the users is needed but at the same time lacks the use of native features that the device offers for native applications. [14.] A hybrid application development approach combines the best of the two approaches by combining the ability to use the devices native features while at the same time using technologies which are used to develop web applications. Hybrid applications have a

very bright future since the use of HTML5 is rapidly growing and the capability is also developing making it very attractive and simpler for developers. The capability of developing an application which can run on multiple platforms is very appealing for anyone who is constrained by financial limitations.

## 5 Technology Implemented

For this thesis project, I have developed a simple hybrid application which can run on multiple platforms such as iOS, Android and windows. It has been tested for the time being on these three platforms and works properly. In this section I will discuss the technologies I have used to develop my hybrid application.

### 5.1 HTML5

According to W3C (World Wide Web Consortium) the world wide web's mark-up language has always been HTML [15]. HTML supports DOM APIs and many tools that support it. Because HTML was developed from the beginning by many people through a long period of time, it has many features that have arisen from those different sources.

HTML5 is the latest advancement in the development of the HTML [16]. The idea of HTML5 denotes two various concepts which describes it very well. The first one is that it has new elements, attributes and behaviours making it the latest version of HTML. The second is that it has and supports larger sets of technologies, which makes it able to allow more various and powerful websites and web applications. [16.] HTML5 has added more mark-up tags than its predecessor HTML4 such as `<header>` and `<footer>` tags to separate the top and bottom of content blocks, `<article>` tag to identify a specific content, `<nav>` tag to identify navigational blocks, `<section>` tag to represent a specific section, `<video>` and `<audio>` tags represent the addition of video and audio content, `<canvas>` tag to support the graphical drawing from other scripts and `<embed>` tag to insert external applications and inputs into the webpage.

Because of HTML5s' improvements from the previous HTML4, its efficiency and improved performances makes HTML5 the future of web browsing. Some of HTML5 benefits include pages will load faster, less bandwidth will be used and energy consumption will be more effective letting devices have longer battery time.

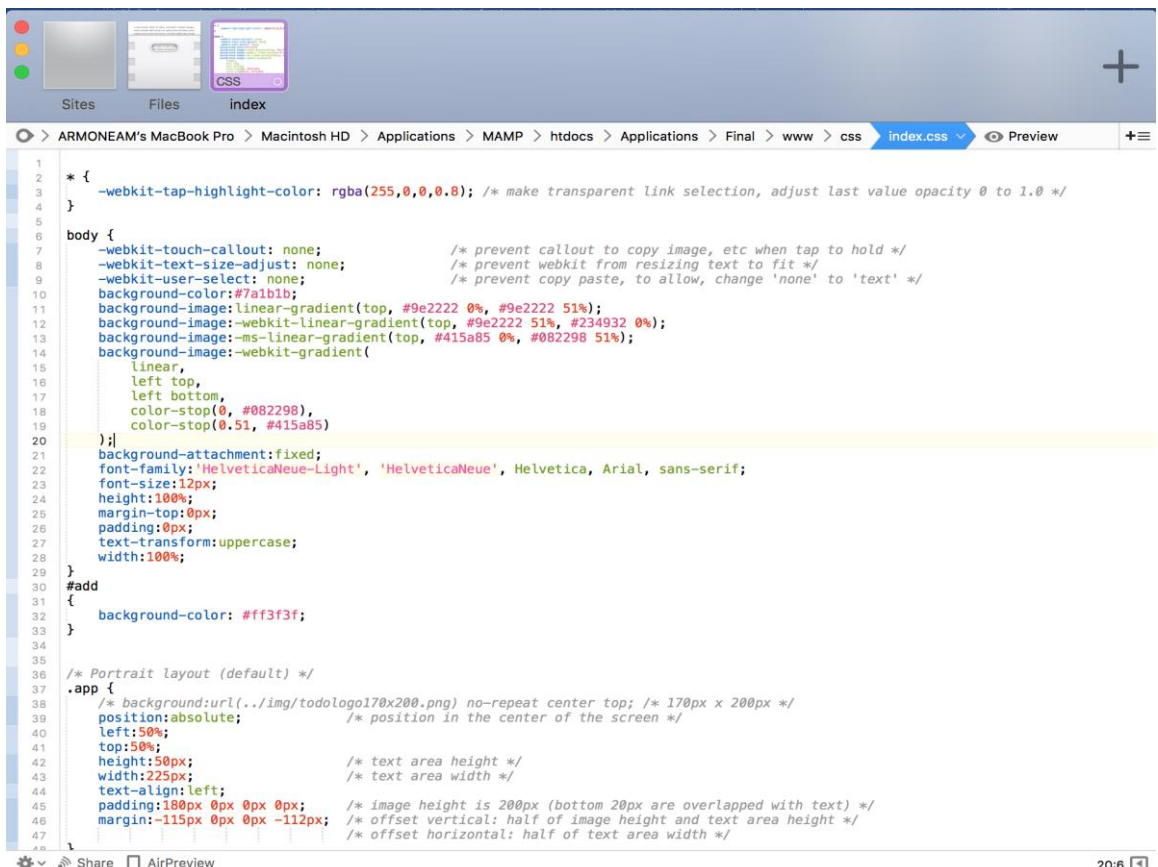
## 5.2 CSS3

CSS3 (Cascading Style Sheet) is the latest version the cascading style sheets formatting language. It is the latest improvement on the previous version (CSS2.1) [17]. It is used to describe the appearance and the formatting of a document such as HTML5.

There are many improvements which are incorporated in the development of CSS3. These include

- Colour
- Selectors Level 3
- Namespaces
- Media Queries

CSS can also be used on documents such as XML (Extensible Mark-up Language), XUL (XML User Interface Language) and SVG (Scalable Vector Graphics) documents making it a versatile and very useful technology.



```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
2559
2560
2561
2562
2563
2564
2565
2566
2567
2568
2569
2570
2571
2572
2573
2574
2575
2576
2577
2578
2579
2580
2581
2582
2583
2584
2585
2586
2587
2588
2589
2590
2591
2592
2593
2594
2595
2596
2597
2598
2599
2600
2601
2602
2603
2604
2
```



tent page which is simple to read and maintain. Another purpose is to increase recyclable codes that can be shared, for example one CSS file can be used by more than one HTML pages. This will help in reducing the complexity of the document and the repetition that might have happened on multiple documents. The CSS specification has a priority scheme to regulate style rules when there are more than one rules which might conflict or stipulate to a single element. [18,7-9.] Other factors which are affected by using CSS file include loading time, lower bandwidth and avoidance of bloated HTML codes.

CSS3 can also render multiple background images, box shadows, text shadows, hover state animations and round corners. To sum up, by using CSS3, we can make documents load faster, use fewer resources, make documents easier to maintain and make them easy to change or modify for the future. Alongside HTML and JavaScript, CSS3 is the foundation of modern day mobile applications, web applications, dynamic and responsive websites that are developed [18,7-9].

### 5.3 Apache Cordova / PhoneGap

Apache Cordova is an open-source mobile application development framework [19]. It allows its users to use web standard technologies such as HTML5, CSS3 and JavaScript for the development of applications. Applications are executed in wrappers which are designed to target specific platforms and use standard APIs to access device's features and capabilities such as camera, vibration, GPS, sensors etc. Apache Cordova is recommended for developers who want to make an application which is capable to run on multiple platforms, or developers who have the ability and knowledge only to develop web applications but want to develop an application which would be available in native application stores, and developers who want to develop applications which are able to mix features of web applications and native applications. [19.]

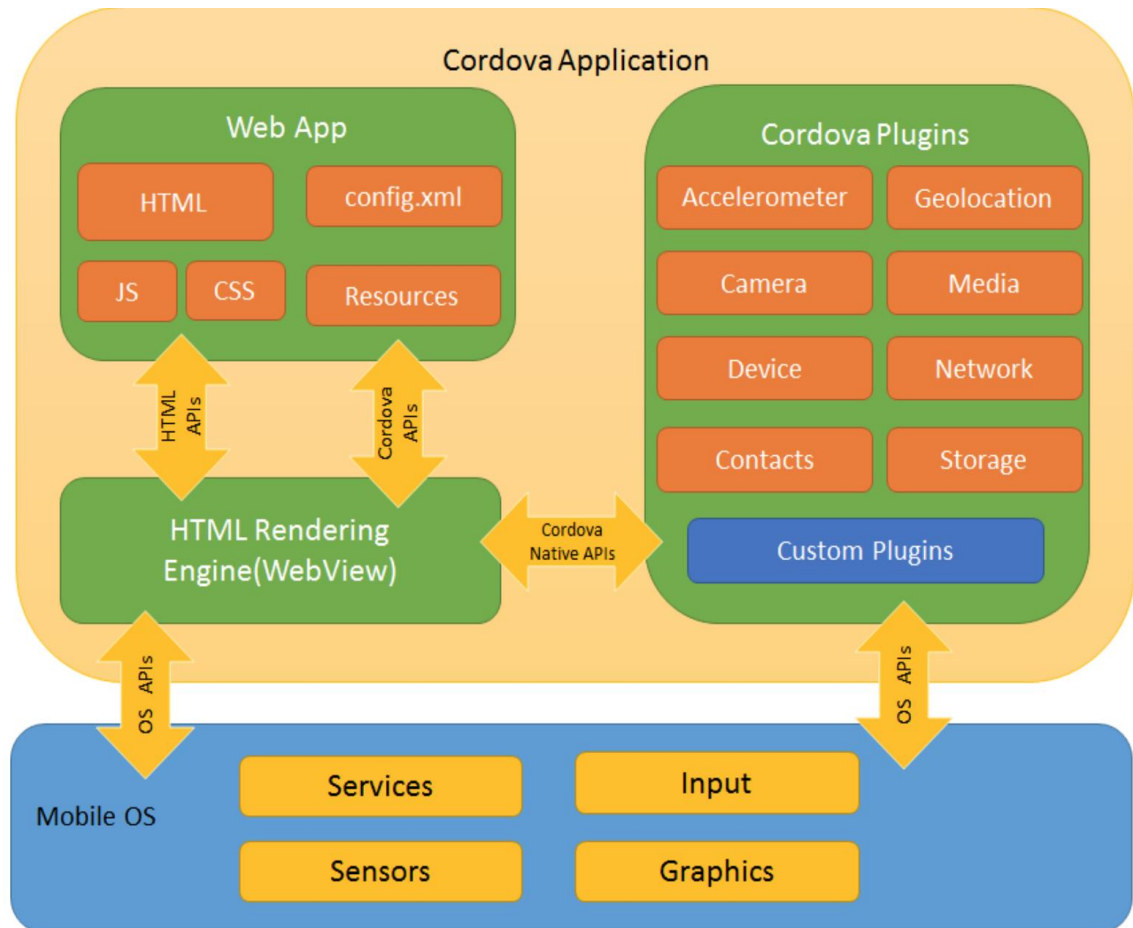


Figure 6. Overview of Cordova/ PhoneGap Application. Reprinted from Apache (2015) [19]

There are several sections of Cordova Application. The above figure shows a view of the Cordova Application Architecture. The application launches in a WebView within a native application wrapper [19]. In this container there is an important file (config.xml), which is responsible for affecting how the application is going to work.

Among important parts of the Cordova system are the plugins [20]. They are the bridge between the Cordova application and the native components to bind together and access to the standard device APIs. The Apache Cordova project team by itself develops and maintains a set of plugins (Core plugins) which are used to give access of device capabilities such as camera, sensors and contacts. to the hybrid application. Even though core plugins are developed by the Apache team, there are many third-party plugins that can be used on applications to access device capabilities.

A basic hybrid application consists of configuration files, Icons for the application and content of the application which is built using web technologies. The configuration file

consists of all the necessary instructions to compile the application. All the contents of configuration file are wrapped in `<widget>` tag. Various options such as full screen, background colour, orientation is declared using `<preferences>` tag. Any application can be designed based on the need on internet access. There are offline applications, online applications and applications which combine both. Offline applications are downloaded entirely on the device and do not need internet access to operate. These applications store offline content on the device so that there is no need of an internet connection to access the necessary content.

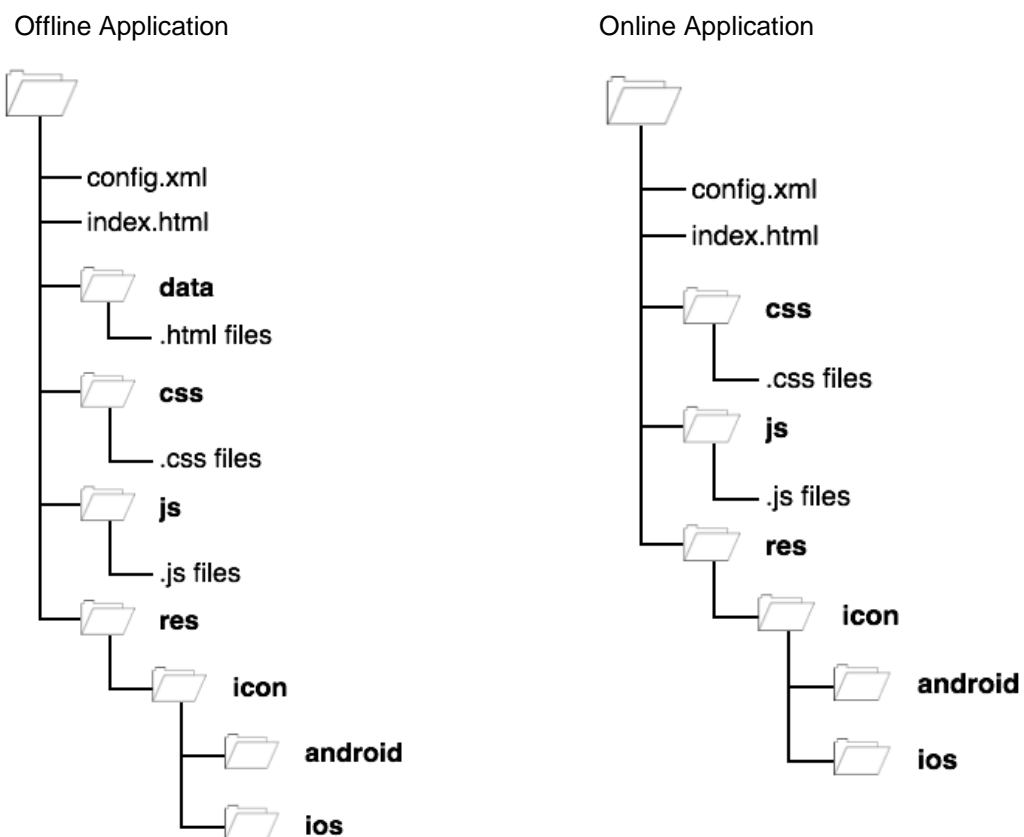


Figure 7. folder structures of offline and online application.

It can be seen that in Figure 7 the data folder is not necessary in Online Apps because all the files and contents needed are stored in remote servers but not on the device. For this thesis project, I have chosen to develop an offline application which will use local storage to store user data.

## 5.4 JavaScript Language and Libraries

JavaScript Language is a high-level programming language which is used to enhance web pages [21,3-4]. The three languages of HTML, CSS and JavaScript create the bases for modern-day web page development. JavaScript is becoming so popular that all modern web browsers support it. Because of this popularity, there is no need to use any plugins when implementing JavaScript on websites. Because JavaScript is already supported by most popular operating systems.

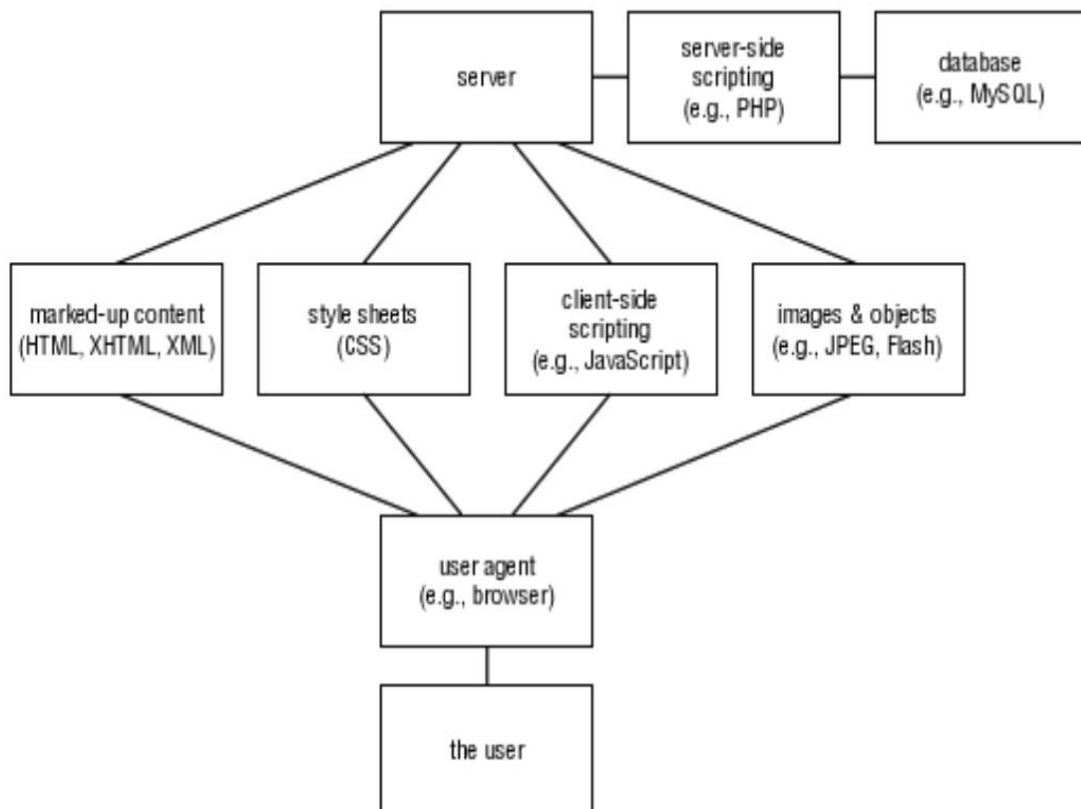


Figure 8. Components of a dynamic website. Reprinted from Goodman (2010) [21,5]

As Figure 8 shows JavaScript works hand in hand with style sheets and marked up content to create the best possible user experience and dynamic web content. Even though JavaScript has many uses and advantages over other programming languages, it has its limitations. As a programmer, it is good to know the advantages and the limitations. JavaScript is better to use when, to get webpages to respond directly to the user, providing small database content is a user-friendly way, controlling different APIs in HTML documents, processing user data before sending to server, dynamic changing of content and style according to user preference and requesting files from server. Ja-

JavaScript limitations include initiating an application on the user's computer, writing or reading files on the user's computer, writing on files which are located on the server, sending secure emails from users to administrators. [21,12-13.] Even if there are some similarities between Java and JavaScript, they are both different programming languages.

#### 5.4.1 JQuery Library

JQuery is a JavaScript library which is small, fast and multi feature which makes HTML handling interesting and easy. It has high adaptability and extensibility [22]. Using JQuery enables easy handling of animations, event handlings and Ajax, JSON parsing and multi browser support.

JQuery library is a single file which is written in JavaScript programming language. There are two ways of adding JQuery library to a project. It is possible to include it either by linking the webpage to a local copy of the JQuery file or it is possible to use a remote copy of the file which can be located on a server anywhere in the world and can be accessed by using https protocol.

```
<script src="jquery.js"></script>
```

Listing 1. Local JQuery file

```
<script src="https://code.jquery.com/  
jquery-3.1.1.min.js">  
</script>
```

Listing 2. Remote JQuery file

While Listing 1 shows the code that is used to state a local JQuery file to be included in a web page, Listing 2 shows the code used to include a remotely located JQuery file. JQuery's Architecture permits the development of plugin codes by developers. Currently there are many plugins created for jQuery library [22]. These plugins are developed for many different purposes and cover different functionalities. They can be easily found on the jQuery project website.

The jQuery Mobile is the touch-optimised version of jQuery library which is currently being developed by the jQuery project team. It is developed based on the core jQuery making it easy to learn for someone who has prior knowledge of jQuery.

## 5.5 Ripple Emulator

A mobile application development process has a completely different path compared to the desktop version. While desktop applications are developed on a device which they are going to be installed, the mobile versions are developed on desk top machines and tested on mobile devices later. This process makes it challenging to test mobile applications side by side on the process of developing them. Because of these challenges, emulators play an important role in the development of mobile application.

Ripple Emulator is one of the most popular emulators which supports different platforms such as Android, iOS and Blackberry. It gives developers the ability to automatically test applications without the need to use a mobile device and restart the emulator, since it supports live automated refresh capability. The Ripple emulator operates as a Google chrome browser extension and therefore there is the need to install a Google chrome web browser to use the Ripple Emulator. [23,45-48.]

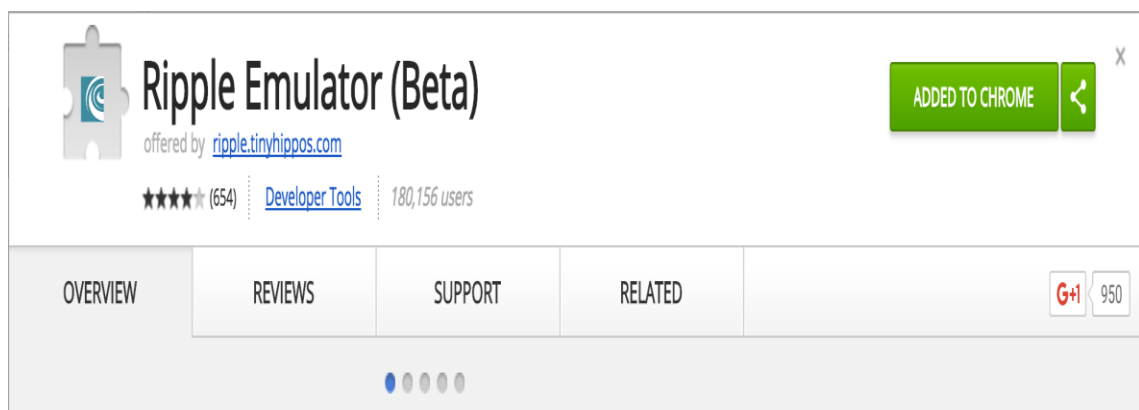


Figure 9. Ripple Emulator extension in a chrome web store.

As Figure 9 shows Ripple Emulator is available in chrome web store's extensions section and can be easily added to the browser as an extension. After it is installed, it will be available by a button which will launch it. In Figure 9, it can be seen how the emulator will be available to users. There will be a button in the right corner of the web browser after installation.

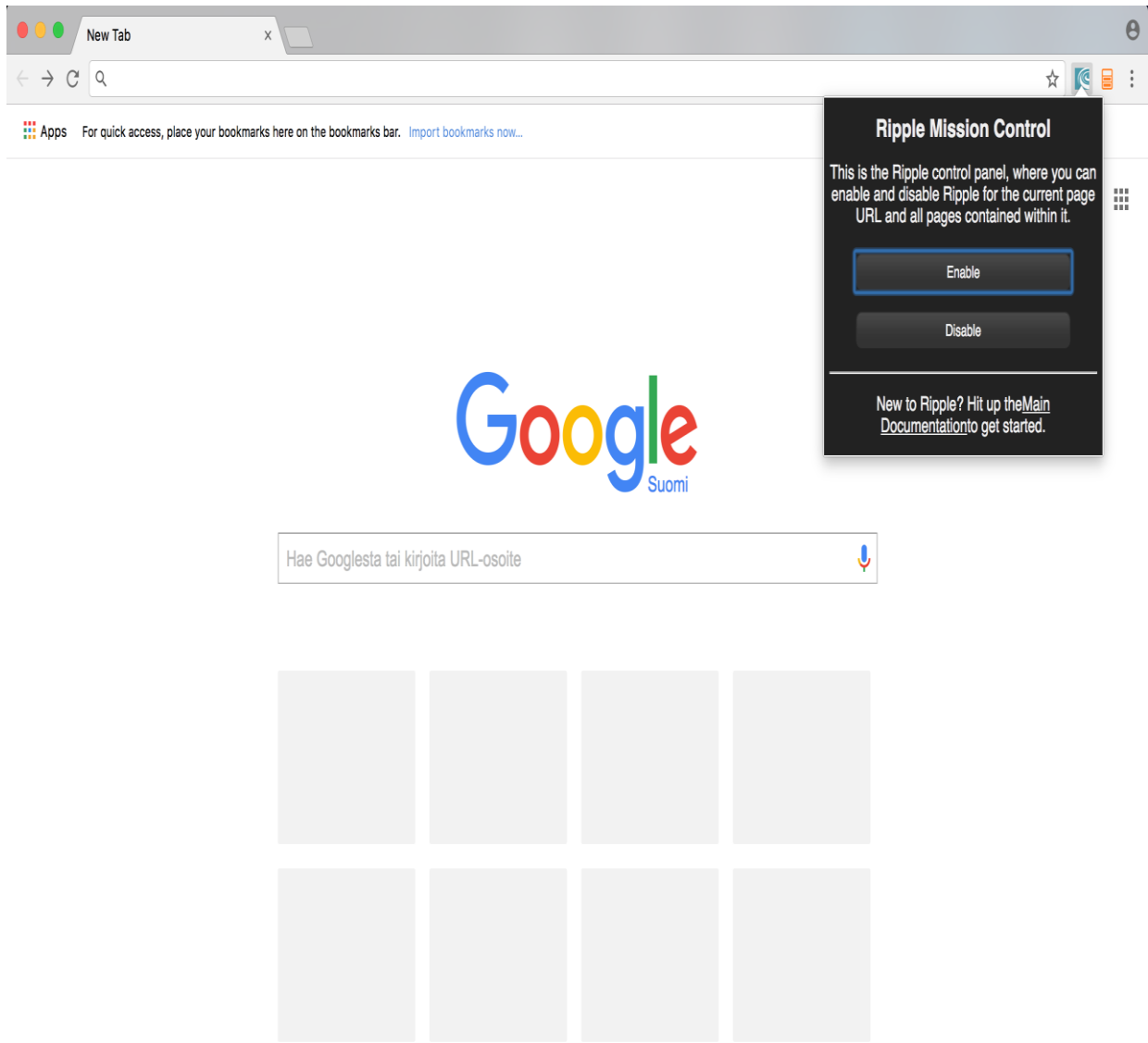


Figure 10. Chrome web browser with installed Ripple Emulator

Figure 10 shows how it is possible to launch the Ripple Emulator. The platform gives different options to choose from, so that it is possible to get the option of available mobile platform frameworks. Figure 11 shows the current available mobile platform frameworks supported by the Ripple Emulator.

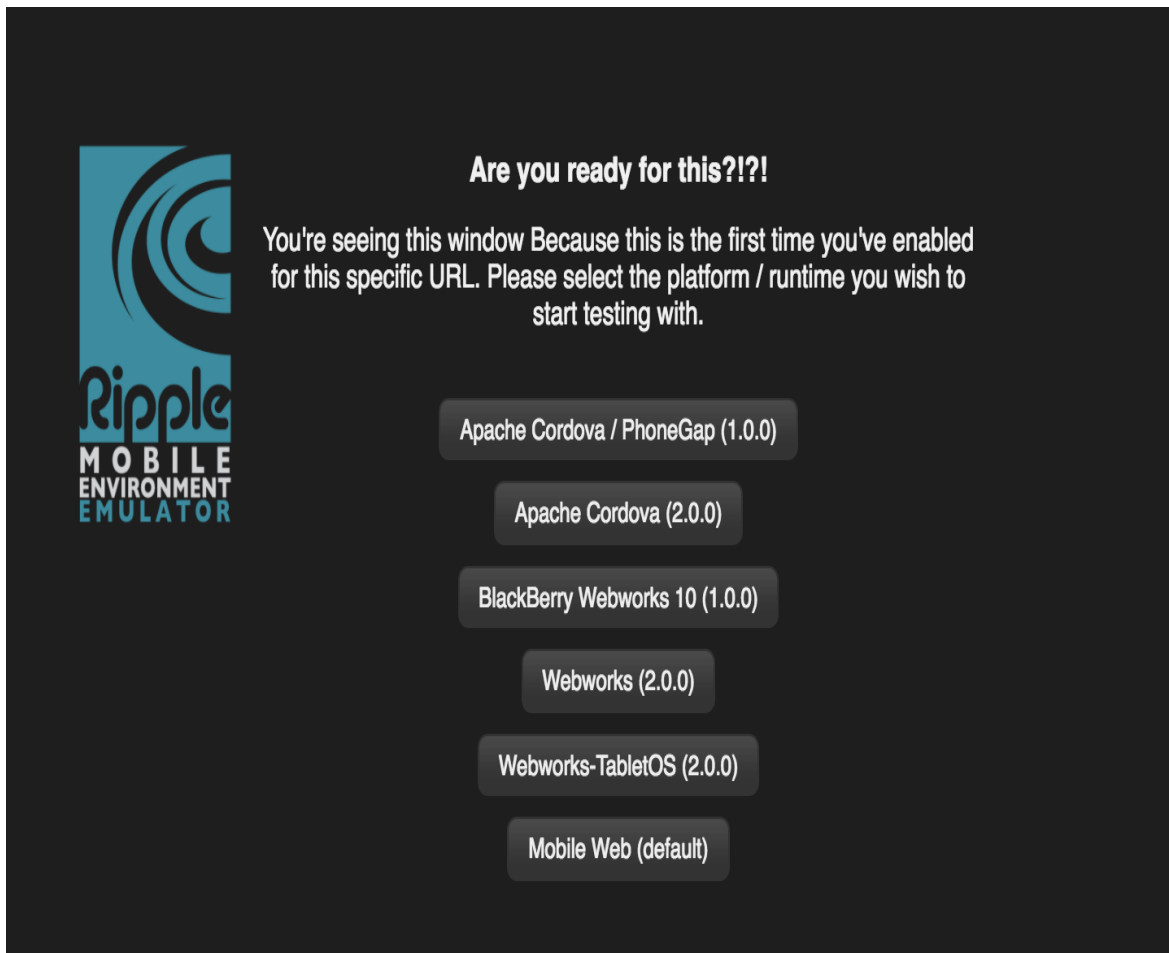


Figure 11. Ripple Emulator start up page

The reason for me to choose the Ripple Emulator was the fact that it supported Apache Cordova / PhoneGap mobile frame work as shown in Figure 11. Ripple has many features such as setting geolocation, shacking the device, device orientation and different screen-size mobile devices. The downside of using emulators is that there is no guarantee that a tested application on the emulator will perform in the same way as it did in the emulator when it was installed on the real device.



## 6 Implementation

In this chapter I will discuss the application that was developed for this thesis project. The name of the application is Bucket List. Its main purpose is to provide a platform for users, where they would be able to save their wishes and keep track of them if they have been achieved or not. In the event of a wish being accomplished, the application will give the chance to take a memento picture and the ability to share that achievement with the rest of the world through either social media, email and text messages.

The first step in this application development was the installation of the PhoneGap (Apache Cordova) framework. Before the installation of PhoneGap, there are two basic requirements that need to be in the development workstation. Node.js and git are prerequisites for the PhoneGap application to work properly. Node.js is a JavaScript runtime environment which is used to build JavaScript code. Git on the other hand is used by a command line interface to download resources.

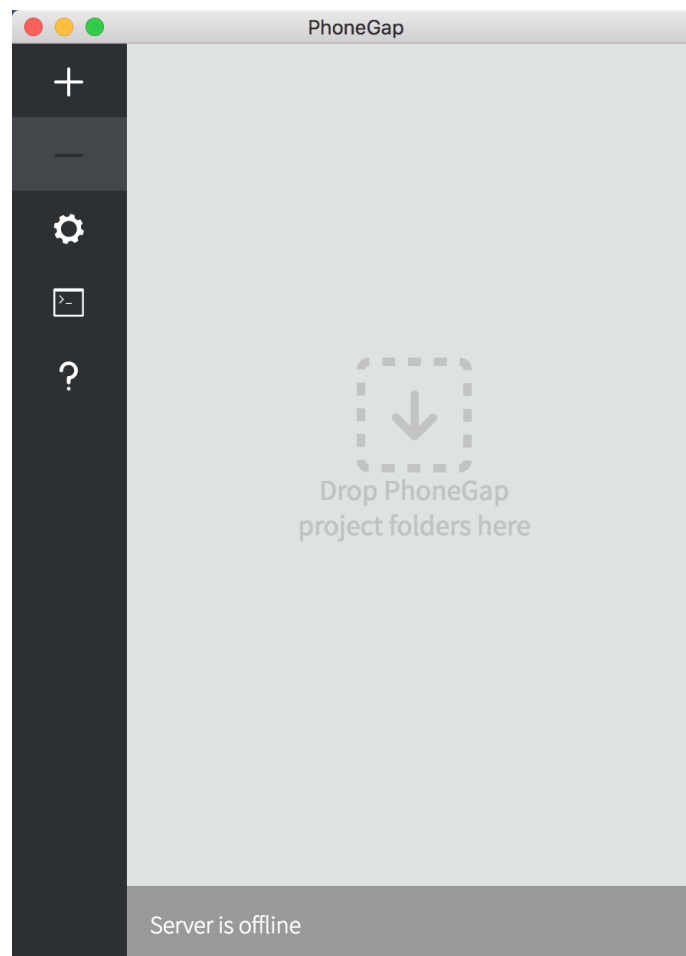


Figure 12. PhoneGap initial page

Figure12 shows the start page of the PhoneGap application which can be used to develop cross-platform applications. After installation, the project can be created by using the 'Create a new PhoneGap project' button. It will automatically suggest a list of simple templets or a blank project for the use of the new project. The next step will be providing the correct pathway for the project and the name of the project.

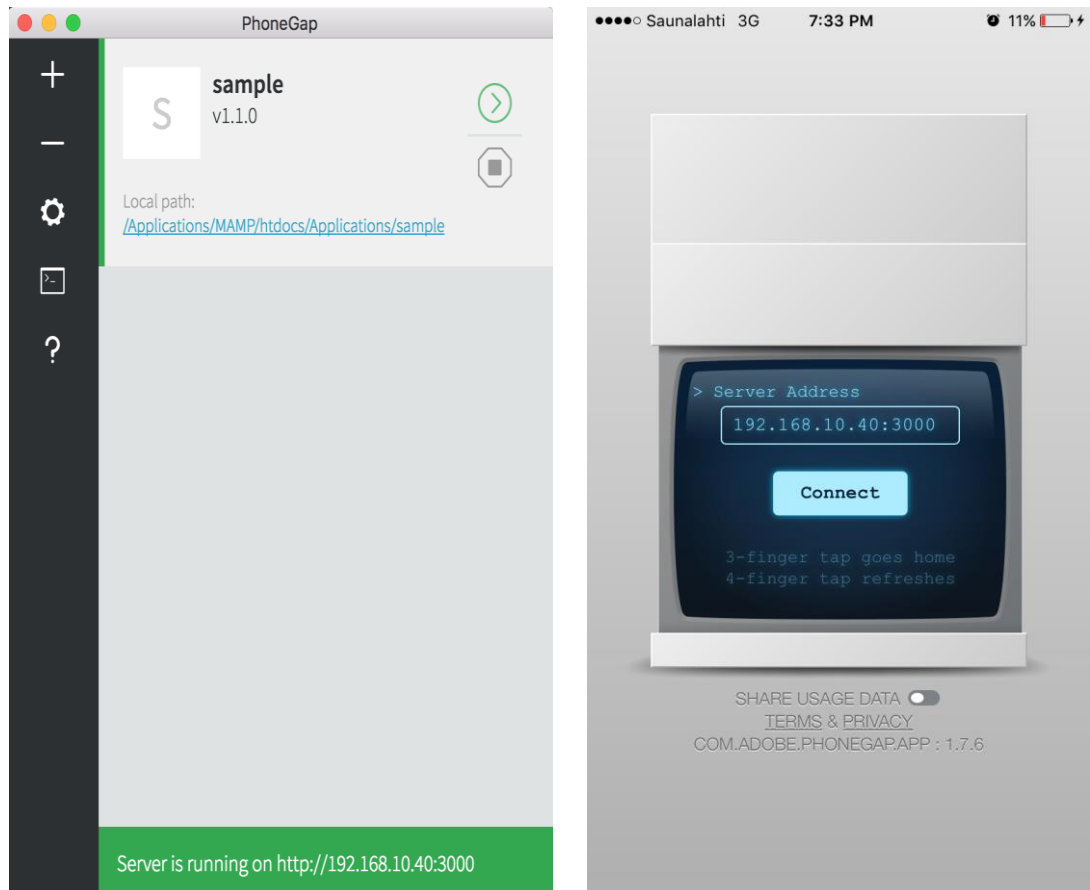


Figure 13. PhoneGap desktop and mobile application versions

Figure 13 shows a sample project created on PhoneGap. It shows the name of the application and the path of the project where the files are located. It also shows that the server on the application is running and the server address which can be used to connect with a device to test the project and shows the PhoneGap application for iOS which can connect with the desktop version enabling live testing of the applications developed without the need to install the application on the device.

## 6.1 Description

As the name of the application describes, the application is used for the purpose of saving users' bucket list of wishes in order for it to be easily saved, edited and easily accessed for referencing. The application has features such as the ability to take a memento picture when a certain wish is accomplished and save it in the pictures gallery or in the camera roll. In addition to this it gives the opportunity to directly share the achievement to social media or directly through email or a text message.

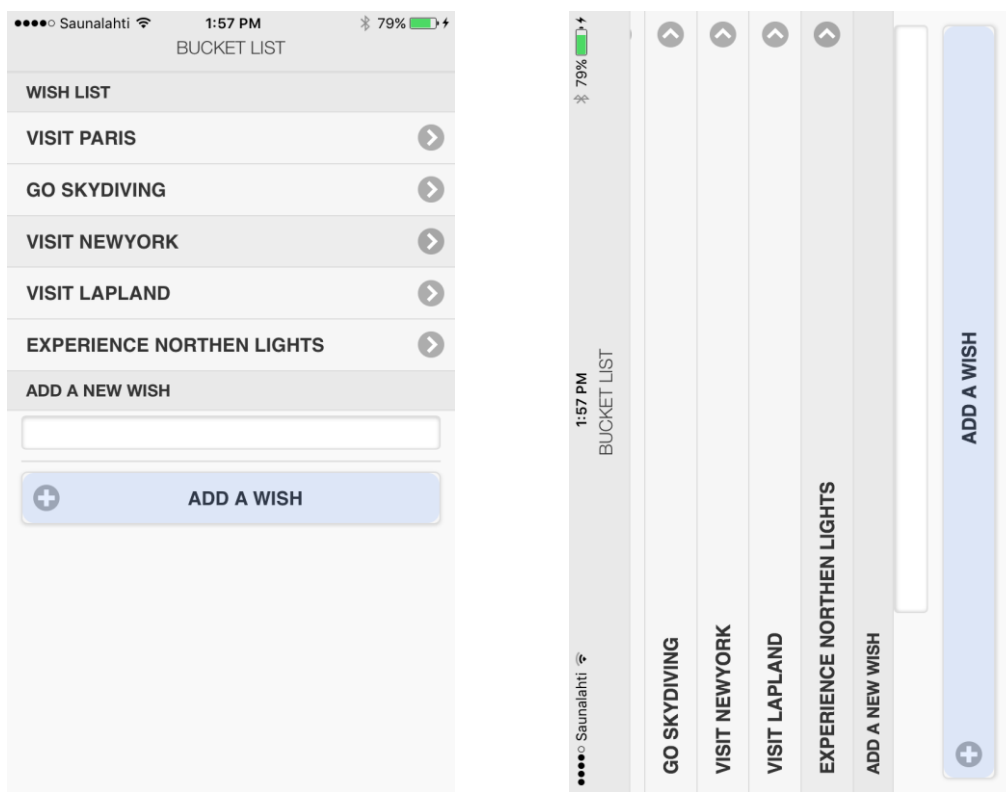


Figure 14. Overview of Bucket List cross platform application

Figure 14 shows the overview of the application developed. The figure which is on the left shows the vertical view of the application while the other one shows the landscape portrayal of the Application. As it can be seen the application is simple and user-friendly. The addition of new wishes is easily achieved by the use of the add a wish button. The user interface is designed by using the HTML code and rendering it using a CSS file.

```

<div data-role="content">
  <ul id="task_list" data-role="listview" >
    <! -- <li data-role="list-divider">WISH LIST</li> -->
  </ul>

  <form id="form" method="GET">
    <div data-role="fieldcontain">
      <label for="inp"></label>
      <input type="text" name="inp" id="inp"/>
    </div>

    <input id="add" type="button" data-icon="plus" value="Add a wish" style="border-radius:10px; background-color: #005aff;"/>
  </form>

</div>

```

Listing 3 Code snippet from HTML file

Listing 3 shows part of the code used to implement part of the front page of the application. It shows how the form was implemented, which will help in the retrieving of data which is stored in the local storage of the device.

## 6.2 Main Components

The Bucket List application has four main components that are displayed to the user so that it will be used for the intended purpose. They are simple and easy to use for any user whether they have been using it for a while or are using it for the very first time. In the following part I will discuss these components in detail.

### 6.2.1 Add a Wish

The Add a Wish feature is used to add new wishes to the bucket list. The added list is stored in the localStorage facility of the device. It is key-value pair database stored locally. In this storage we can store any string and that data will be available on that de-

vice later on. The downside of this method is that the data stored on this device will not be available from any other device. For this application, I have used a single key in the local storage and the value will be a stringified version of the wish list items that are stored.

```
wish.add = function(event) {
  // Read the task from the input
  var wish=$('#input').val();
  if (wish) {
    // Add the task to array and refresh list
    wish.list[wish.list.length] = wish;
    wish.refresh_list();
    // Clear the input
    $('#input').val('');
  }
  event.preventDefault();
};

-----

// Store back the list
localStorage.todo_list = JSON.stringify(todo.list ||
[]);

-----

// Load the list by parsing the JSON from localStorage
todo.list = JSON.parse(localStorage.todo_list);
$('#add').bind('vclick', todo.add);
$('#task_list').on('vclick', 'li a', function() {
  todo.selected = $(this).data('task');
});
```

Listing 4 Code snippet from wish add function

Listing 4 code snippet shows the way how a new wish was added and the method used to store the added wish. `JSON.parse` was used to convert the JSON string back to JavaScript data and return that to the front page of the HTML.

## 6.2.2 Wish List Update

After exploring the Add a Wish feature comes the Wish List Update Page. Its use is simple and easy for the user. It is accessed by clicking on the arrow sign which is found on every wish list component. It gives the choice of updating the wish list by marking the selected wish as either Achieved or Not Achieved. It also gives the option of removing the wish completely from the wish list.

When the user updates a wish to be achieved, then the application will give a congratulatory message and an option to take a memento picture to remember the achievement or to share it to others. After this the wish will be removed from the wish list and the updated list will load on the front page of the application. In the event the user clicks on the Not Achieved button, the application will give an encouragement message and keep the wish in the wish list.

```

<div id="confirm" data-url="confirm" data-role="page">
<div data-role="header">
<h1>WISH LIST UPDATE</h1>
</div>
<div data-role="content">
Mark this Wish as<br>
<a class="remove_task" href="#done" data-role="button"
data-icon="check" data-theme="f">Achieved</a>
<a href="#notdone" data-role="button" data-
icon="delete" data-theme="g">Not Achieved</a>
<a class="remove_task" href="#index" data-
role="button" data-icon="alert" data-theme="f">Remove
Wish</a>
<br><br>
<a href="#index" data-role="button" data-
icon="minus">Cancel</a>
</div>
</div>

```

Listing 5 Code snippet of Wish List update

Listing 5 demonstrates the code used to implement the Wish List Update page of the application. In case of an accidental click or change of mind by the user, there is a button for cancelling and it goes back to the main page without the need to update the wish list.

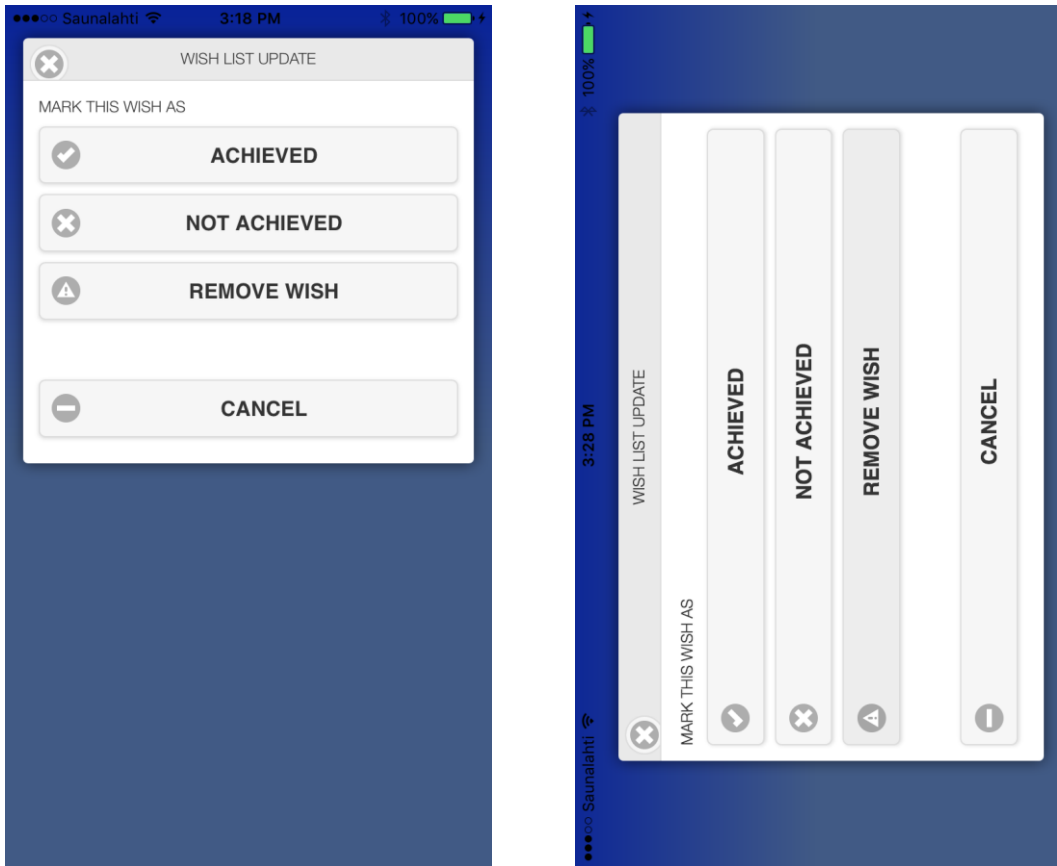


Figure 15 Wish update list page

The screen shot picture of the wish update list page is demonstrated in Figure 15. It can be seen that in both orientations the user interface remains simple and attractive for users. A gradient colour background was used to give the darker blue colour to the top and the lighter blue colour to the bottom of the application.

### 6.2.3 Take a Picture

The Take a Picture feature is used to take a souvenir picture to mark the moment or to share it with others. One of the core plugins for PhoneGap framework is a camera plugin. It gives access to hybrid apps to the device's camera features.

The plugin is added by using a command line interface directly or via repo URL. It can be added by the following command.

```
$cordova plugin add cordova-plugin-camera
```

or

```
$cordova plugin add https://github.com/apache/cordova-plugin-camera.git
```

Listing 6 Cordova camera plugin installation

Listing 6 shows the command line that can be used in the installation of the Cordova camera plugin. After the installation, the implementation was performed by using the codes shown in Listing 7.

```
capturePhoto: function(){
    navigator.camera.getPicture(onSuccess, onFail, { quality: 50,
    destinationType: Camera.DestinationType.DATA_URL,
    saveToPhotoAlbum:true });

    function onSuccess(imageData) {
        var image = document.getElementById('image');
        image.style.display="block";
        image.src = "data:image/jpeg;base64," + imageData;
    }
    function onFail(message) {
        alert('' + message);
    }
}
```

Listing 7 Cordova camera plugin installation

As Listing 7 shows, the camera.getPicture function opens the device's camera for it to be ready to take a picture. Listing 7 shows that I have used the Base64 encoding of the image data and the image taken to be saved in the photo album by using the saveToPhotoAlbum:true parameter.



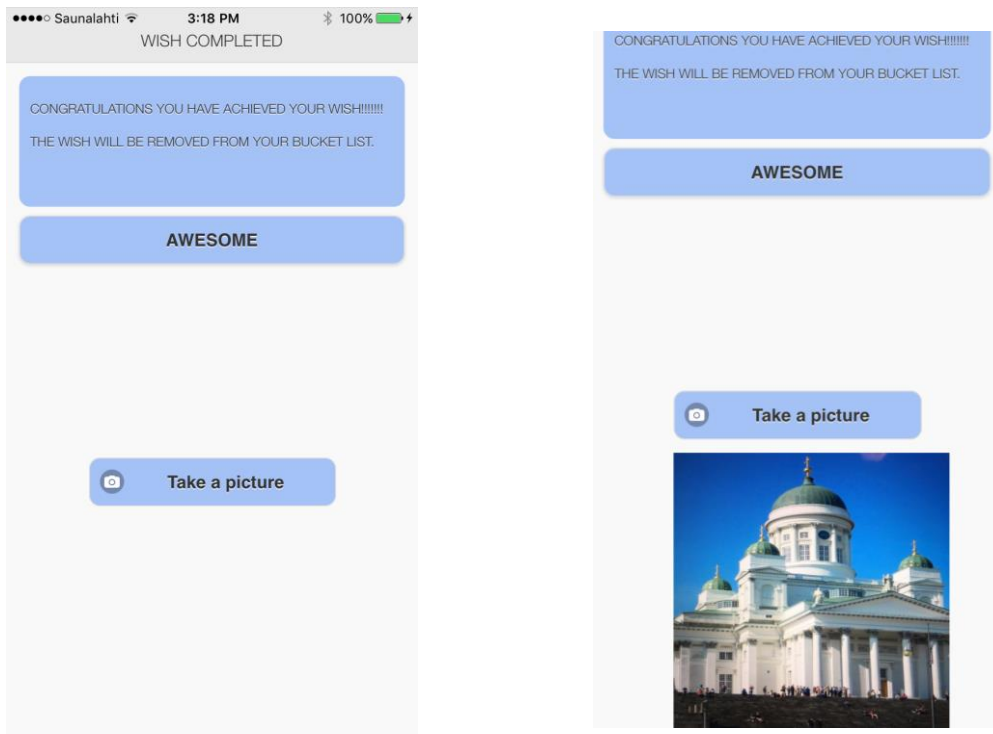


Figure 16 Take a Picture feature button

Figure 16 shows the before a picture is taken and after use of the take a picture feature. The left one shows before the souvenir picture was taken. On the right side, it shows how the picture that was taken will be shown to the user.

```

<div >
  <p> <button onclick="app.capturePhoto()" class="ui-btn
  ui-icon-camera ui-btn-icon-left" style="border-radius:
  10px;background-color:#a5c2f7;">
  Take a picture</button></p>
  <div>
   </div>
  </div>

```

Listing 8 code snippet for camera and picture display

Listing 8 shows the code implementation of call for capturePhoto function and the image display section. <img> tag is used to determine the size of the thumbnail picture on the screen.

## 6.2.4 Social Sharing

The Social sharing plugin for this project was used from a third-party plugin developer. The social share plugin allows the use of native social sharing window of the mobile device. The plugin was installed by using the following command line interface command.

```

window.plugins.socialsharing.share(message,
subject,file,url,[successCallback],
{console.log('result: ' + result)}
[errorCallback]
{alert('error: ' + result);
});

```

Listing 9 Native social share plugin [24]

As Listing 9 shows it is possible to share a different combination of data from the application. Even though there are different possibility options, the only necessary for my application was to share an image accompanied by a message. The other parameters are not necessary for this project. Figure 17 shows the application using the native share sheet.

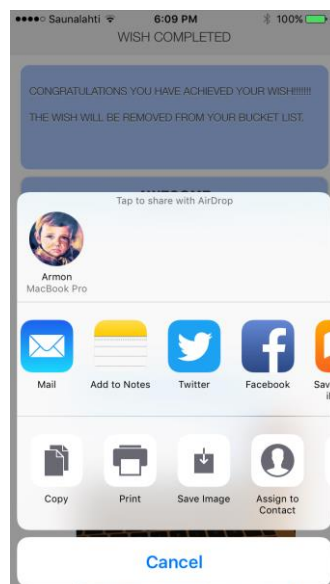


Figure 17 Native social share sheet

As Figure 17 shows I have used the native social share sheet instead of a plugin developed by a third-party. In PhoneGap plugin archives there are many social share plugins developed by third parties.

## 7 Testing

Mobile application development is considered very challenging amongst many developers. Mobile environment and devices bring more challenges compared to the actual development of the application itself. [25,213.] In order to check the functionality of my application it was tested on two environments; on a real phone and in a simulator or emulator environment.

### 7.1 Testing Using Emulator

Testing using an emulator is easy and gives different options to test different features. For this project, as I have mentioned in earlier chapters, I have used the Ripple Emulator to test the application. Ripple Emulator supports different screen sizes and different operating system mobile devices.

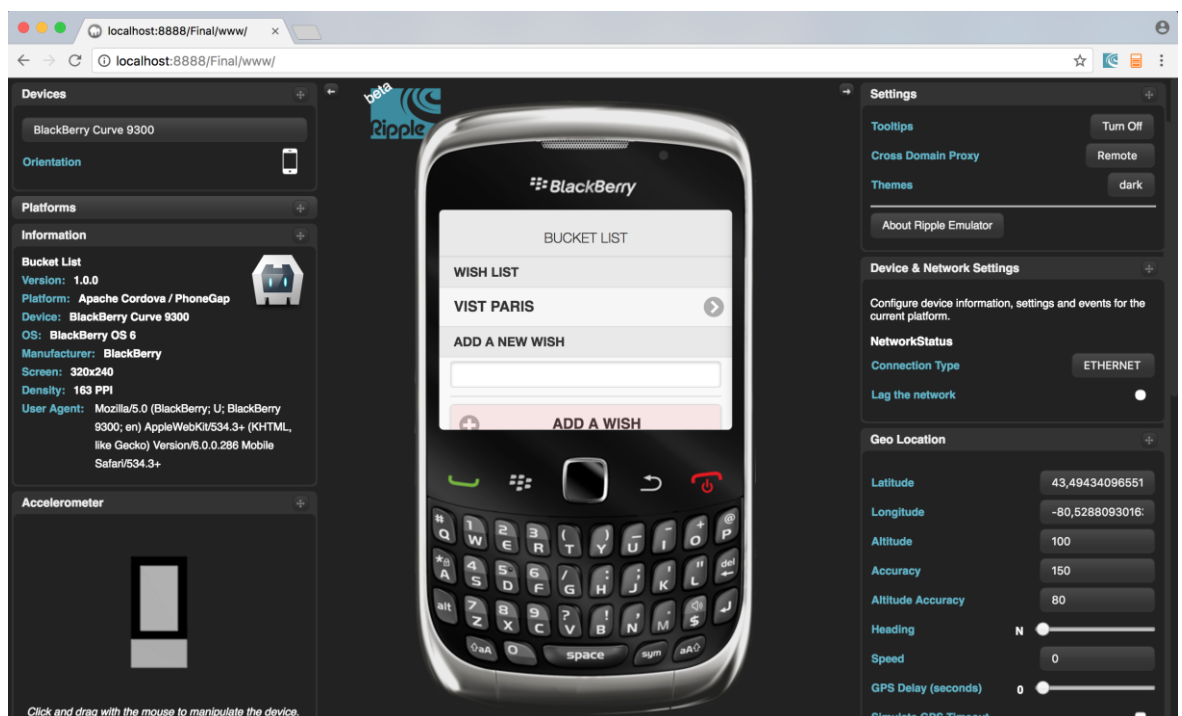


Figure 18 Ripple Emulator Application Testing

As Figure 18 shows the application worked as expected on the many platforms available that it was tested on. It was possible to test it on operating systems such as iOS, Android, Windows and BlackBerry. Even though testing on an emulator is easy, it has

drawbacks for testing. For example, it was not possible to test the camera feature of the application on the emulator because it does not support that feature.

## 7.2 Testing Using a Real Phone

To test the application on a real device the application must be built into installation package by Adobe PhoneGap build.

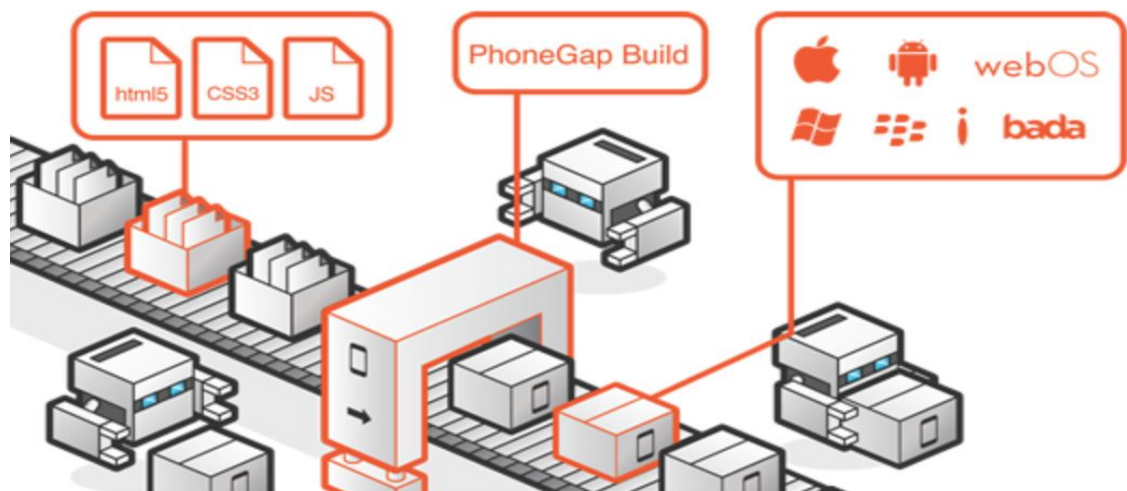


Figure 19 PhoneGap Build. Reprinted from Apache (2015) [19]

As Figure 19 shows an application developed by using HTML, CSS and JS is built into an installation package which can be installed on different operating systems. After this process, the application can be installed by downloading on the test device or by a simple scan of a QR code of the app which will automatically install the application.

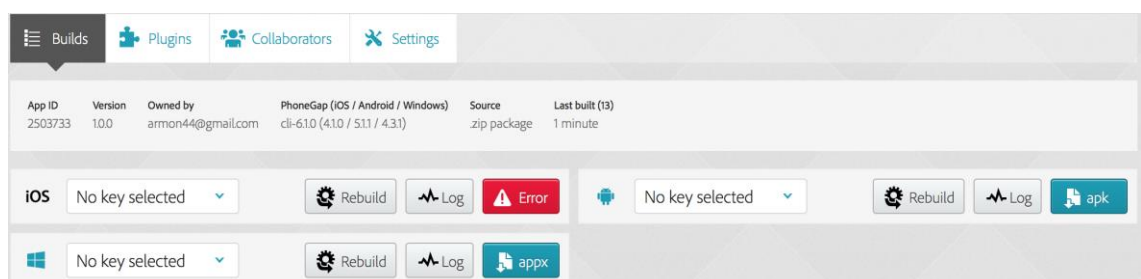


Figure 20 PhoneGap Build result

The results of the build process from the application can be seen from Figure 20. The apk file for Android and the appx file for a Windows mobile phone are ready to be installed or published in their respective application stores. When it comes to the iOS installation file it shows an error because I was not a subscribed Apple developer so I could not provide a signing certificate and keychain pair in my application.

### 7.2.1 iOS Device

Because I do not have an Apple developer account, it was not possible to directly install the application on an iOS device but it was possible to use a PhoneGap developer application for iOS, which will allow to run the application on an iOS device. On this test the Application performed as expected and all features were working. Most of the screen shots in the above chapters were taken from the application working on an iOS device. The device I used was iPhone 6s with iOS version 10.2.1.

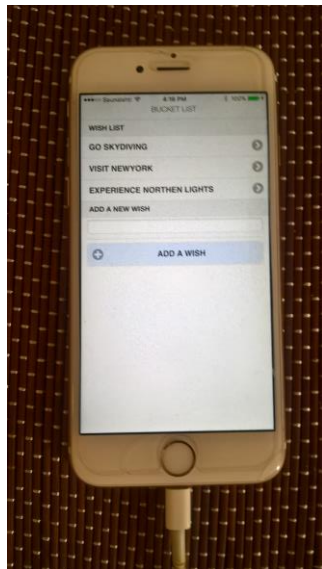


Figure 21 Bucket List Application on a iOS mobile device.

Figure 21 was taken as a demonstration of Bucket List running on an iOS device. It fitted the screen properly and the fonts were rendered correctly.

### 7.2.2 Windows Mobile Device

With a Windows mobile phone there was the same problem as in the case of iOS and it was not possible to install the application without a developer account. Although I was not able to install it, I was able to use a PhoneGap developer application to run the app on a Windows platform. On a Windows phone the application was tested and per-

formed as it should without any major issue. The device I used was Microsoft Lumia 650 with a Windows 10 mobile operating system.

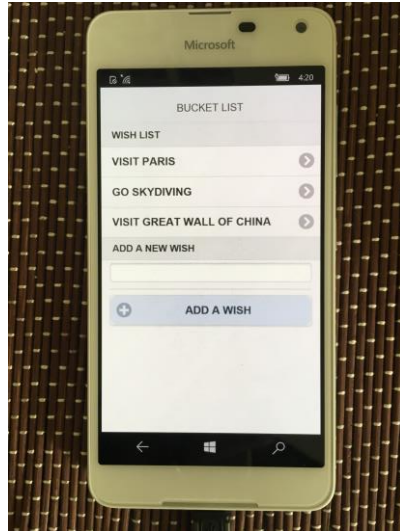


Figure 22 Bucket List Application on a windows mobile device.

As Figure 22 shows the application was performing as expected. There was some speed issue detected but it was not that significant.

### 7.2.3 Android Device

Installation on an Android device was the easiest from the three devices chosen for this project. Since the operating system is open-source software it was possible to install the Bucket List Application without any problem. All it took was the QR code from PhoneGap build and then the application downloads and installs automatically. The device I used was Samsung Galaxy Ace 3 with Android version 4.2.2.

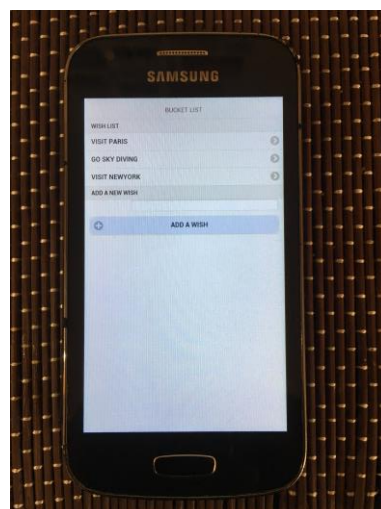


Figure 23 Bucket List Application on an Android mobile device.

From this test, as can be seen from Figure 23, it was evident that the font size was very small compared to what I had expected. Other than the font issue, the application and all its features were performing as expected.

## **8 Result and Discussion**

In this section I will discuss the positive outcomes I gained from this project, challenges I faced and finally improvements that can be done to this application.

### **8.1 Positive Outcomes**

The result of this project was the development of the Bucket List Application using the hybrid mobile application development method. The project was a successful one and it has helped me to know and to familiarize myself with the application development process. Using the PhoneGap development framework has helped me in making the development process easy and getting a cross-platform application without the need to write the code for each platform. I have come to understand the future is bright when it comes to HTML5's versatility and powerful features, making it very attractive for developers around the world. The benefit of cross-platform applications is not only for developers, but also for companies who are concerned about the cost of application development for many platforms and maintaining them.

### **8.2 Challenges**

During this project everything was not rosy and easy. I had to face a few challenges. One of the biggest challenges that I faced was familiarizing myself with PhoneGap framework. It was a new framework for me so it was challenging to study it by myself and examine how it works. Other challenges include lack of real devices to test my application on, not having a developer account for Windows mobile and iOS, dependency on an emulator to test for many devices and screen sizes, outdated information on the internet, lack of user feedback since I was not able to publish my application in app stores yet. Those were some of the challenges I faced during this project.

### 8.3 Future Improvements

The application that was developed is not a final product but rather a big step towards a fully functioning robust application. Some of the planned improvements were noticed during the developing process while the rest were planned after the completion of this project.

- Online storage feature since the application stores locally on the device
- User login and access of saved items from different devices
- Better user interface
- Storage of achieved wishes with accompanying pictures in the application rather than saving the picture in the photo album of the device
- Taking and saving video after achieving a wish



## 9 Conclusion

From the beginning the aim of this thesis project was to research different approaches of mobile device application development and make an informed choice based on merit. This project was a chance to brush up and use my HTML5, CSS3 and JavaScript knowledge and skills. Knowing a programming language is different from knowing how it can work together with another one to produce an application. Furthermore, I have found out that a hybrid mobile application is the most versatile and practical choice since it was possible to target a wide range of mobile platforms. Even though hybrid applications may not give high performance compared to native applications because of the extra layer between the code and the targeted mobile platform, they compensate it by the reduction in development time and cost.

In a rapidly growing and changing technological field it was a challenge to get up-to-date information. Some of the information that I found and thought would be helpful was outdated and old or is not supported any more. It is easy to see how fast things are changing and the use of a cross platform application is being adopted by big platform developers such as Microsoft.

To summarize my experience in this project I was able to create a working hybrid application capable of running on multiple platforms. I have tested the application on three different devices from three different platforms. In addition to that, I was able to test it on the Ripple Emulator to see how it performs on different screen sizes and platforms. The results from these different tests are very satisfactory and encouraging for future endeavours.

## References

1. Hoehle H, Venkatesh V. Mobile Application Usability: Conceptualization and instrument development; 2015.
2. Meeker M. Internet Trends 2016 – Code Conference [online]. Kleiner Perkins Caufield & Byers; June 2016.  
URL: <http://www.kpcb.com/internet-trends>. Accessed 25 December 2016.
3. Open Handset Alliance. Industry leaders announce open platform for mobile devices [online]. Open Handset Alliance; November 2007.  
URL: [http://www.openhandsetalliance.com/press\\_110507.html](http://www.openhandsetalliance.com/press_110507.html). Accessed 26 December 2016.
4. Android open source project. Android, the world's most popular mobile platform [online]. Android open source project.  
URL: <https://developer.android.com/about/android.html>. Accessed 29 December 2016.
5. Apple Inc. 27<sup>th</sup> WWDC. Platforms state of the union [online]. San Francisco, CA; Apple Inc.  
URL: <https://developer.apple.com/videos/wwdc2016/>  
Accessed 10 January 2016.
6. McWherter J, Gowell S. Professional Mobile Application Development. Somerset: John Wiley & Sons, Incorporated; 2012.
7. Hoffman C. World Beyond Windows [online]. Pc World; September 2016.  
URL: <http://www.pcworld.com/article/3124563/open-source-tools/mozilla-is-stopping-all-commercial-development-on-firefox-os.html>  
Accessed 5 January 2016.
8. Lunden I. Mozilla Will Stop Developing and Selling Firefox OS Smartphones [online]. Tech Crunch; December 2015.  
URL: <https://techcrunch.com/2015/12/08/mozilla-will-stop-developing-and-selling-firefox-os-smartphones/?ncid=rss>. Accessed 5 January 2016.
9. Jolla Oy. The birth of sailfish operating system [online]. Jolla Oy.  
URL: <https://sailfishos.org/about/>. Accessed 12 January 2017.
10. Ubuntu Mobile developer. Ubuntu phone development: Get started [online]. Canonical Ltd.  
URL: <https://developer.ubuntu.com/en/phone/> Accessed 12 January 2017.
11. Nielsen J, Budiu R. Mobile Usability. Berkley, CA: New Riders; 2013.
12. Malavolta I. Beyond native apps: web technologies to the rescue. Mobile! 2016; keynote: 1-2.

13. International Business Machines (IBM). Native, web or hybrid mobile development[online]. Somers, NY: IBM corporation; April 2012. URL: <ftp://public.dhe.ibm.com/software/pdf/mobile-enterprise/WSW14182USEN.pdf>. Accessed 12 January 2017.
14. Xanthopoulos S, Xinogalos S. Mobile App Development in HTML5. AIP Conference Proceedings 2015; 1648(1): 1-4.
15. World wide web consortium (W3C). HTML5: A vocabulary and associated APIs for HTML and XHTML [online]. W3C; October 2014. URL: <https://www.w3.org/TR/html5/introduction.html#background>. Accessed 15 January 2017.
16. Mozilla Developer Network. HTML5 [online]. Mozilla Developer Network and individual contributors; September 2016. URL: <https://developer.mozilla.org/en/docs/Web/Guide/HTML/HTML5>. Accessed 17 January 2017.
17. Mozilla Developer Network. HTML5 [online]. Mozilla Developer Network and individual contributors; September 2016. URL: <https://developer.mozilla.org/en/docs/Web/CSS/CSS3>. Accessed 18 January 2017.
18. Frain B. Responsive Web Design with HTML5 and CSS3. Olton, TX: Packt Publishing; 2012.
19. The Apache Cordova. Apache cordova overview [online]. The Apache Cordova Software foundation; 2015. URL: <https://cordova.apache.org/docs/en/latest/guide/overview/>. Accessed 20 December 2016.
20. Natili G. PhoneGap Beginner's Guide. Birmingham, UK: Packt Publishing; 2013.
21. Goodman D, Morrison M, Novitski P, Rayl T, Gustaff R, Tia G. JavaScript Bible. Hoboken, NJ: John Wiley & Sons Incorporated; 2010.
22. jQuery. What is jQuery [online]. The jQuery Foundation; 2017. URL: <https://jquery.com>. Accessed 21 December 2016.
23. Nair NB, Ragini K. Building Mobile Applications Using Kendo UI Mobile and ASP.NET Web API. Olton, TX: Packt Publishing; 2013.
24. Progress Telerik. Social sharing Phone Gap plugin [online]. Progress Software Corporation; 2017. URL: <http://plugins.telerik.com/cordova/plugin/socialsharing>. Accessed 29 January 2017.
25. Myers GJ, Sandler C, Badgett, T. The Art of Software Testing. Hoboken, NJ: John Wiley & Sons Incorporated; 2011.

## Appendix 1: Code Listing from config.xml

```
1 <?xml version='1.0' encoding='utf-8'?>
2 <widget id="Bucket_List" version="1.0.0" xmlns="http://www.w3.org/ns/widgets" xmlns:gap="http://phonegap.com/ns/1.0">
3   <name>Bucket List</name>
4   <description>
5     Bucket List
6   </description>
7   <author email="armon44@gmail.com" href="http://yourdomain.com">
8
9
10    </author>
11    <preference name="permissions" value="none" />
12    <preference name="phonegap-version" value="cli-6.1.0" />
13    <preference name="orientation" value="default" />
14    <preference name="target-device" value="universal" />
15    <preference name="fullscreen" value="true" />
16    <preference name="webviewbounce" value="true" />
17    <preference name="prerendered-icon" value="true" />
18    <preference name="stay-in-webview" value="false" />
19    <preference name="ios-statusbarstyle" value="black-opaque" />
20    <preference name="detect-data-types" value="true" />
21    <preference name="exit-on-suspend" value="false" />
22    <preference name="show-splash-screen-spinner" value="true" />
23    <preference name="auto-hide-splash-screen" value="true" />
24    <preference name="disable-cursor" value="false" />
25    <preference name="android-minSdkVersion" value="7" />
26    <preference name="android-installLocation" value="auto" />
27    <gap:plugin name="org.apache.cordova.battery-status" source="npm"/>
28    <gap:plugin name="org.apache.cordova.camera" source="npm" />
29    <gap:plugin name="org.apache.cordova.media-capture" source="npm" />
30    <gap:plugin name="org.apache.cordova.console" source="npm" />
31    <gap:plugin name="org.apache.cordova.contacts" source="npm"/>
32    <gap:plugin name="org.apache.cordova.device" source="npm"/>
33    <gap:plugin name="org.apache.cordova.device-motion" source="npm" />
34    <gap:plugin name="org.apache.cordova.device-orientation" source="npm"/>
35    <gap:plugin name="org.apache.cordova.dialogs" source="npm"/>
36    <gap:plugin name="org.apache.cordova.file" source="npm" />
37    <gap:plugin name="org.apache.cordova.file-transfer" source="npm"/>
38    <gap:plugin name="org.apache.cordova.geolocation" source="npm"/>
39    <gap:plugin name="org.apache.cordova.globalization" source="npm"/>
40    <gap:plugin name="org.apache.cordova.inappbrowser" source="npm"/>
41    <gap:plugin name="org.apache.cordova.media" source="npm"/>
42    <gap:plugin name="org.apache.cordova.network-information" source="npm"/>
43    <gap:plugin name="org.apache.cordova.splashscreen" source="npm"/>
44    <gap:plugin name="org.apache.cordova.vibration" source="npm"/>
45    <icon src="icon.png" />
46    <icon gap:platform="android" gap:qualifier="ldpi" src="res/icon/android/icon-36-ldpi.png" />
47    <icon gap:platform="android" gap:qualifier="mdpi" src="res/icon/android/icon-48-mdpi.png" />
48    <icon gap:platform="android" gap:qualifier="hdpi" src="res/icon/android/icon-72-hdpi.png" />
49    <icon gap:platform="android" gap:qualifier="xhdpi" src="res/icon/android/icon-96-xhdpi.png" />
```

**Appendix 2: Code Listing from index.html**

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <meta charset="utf-8" />
5   <meta name="format-detection" content="telephone=no" />
6   <meta name="msapplication-tap-highlight" content="no" />
7   <meta name="viewport" content="user-scalable=no, initial-scale=1, maximum-scale=1, minimum-scale=1, width=device-width, height=device-height, target-densitydpi=device-dpi" />
8
9   <title>Bucket List</title>
10  <link rel="stylesheet" type="text/css" href="css/jquery.mobile-1.4.3.min.css">
11  <link rel="stylesheet" type="text/css" href="css/index.css" />
12  <link rel="stylesheet" type="text/css" href="css/jquery.mobile-1.0.1.custom.css?" />
13  <script type="text/javascript" src="js/jquery-1.11.1.min.js"></script>
14  <script type="text/javascript" src="js/jquery.mobile-1.4.3.min.js"></script>
15
16 </head>
17
18 <body>
19   <script type="text/javascript" src="cordova.js"></script>
20   <script type="text/javascript" src="js/todo.js"></script>
21   <div id="index" data-url="index" data-role="page">
22     <div data-role="header" data-position="fixed">
23       <span> <h1>Bucket List</h1> </span>
24     </div>
25
26     <div data-role="content">
27
28       <ul id="task_list" data-role="listview" >
29         <!-- <li data-role="list-divider">WISH LIST</li -->
30       </ul>
31
32       <form id="form_336" method="GET">
33
34         <div data-role="fieldcontain">
35           <label for="inp_337"></label>
36           <input type="text" name="inp_337" id="inp_337" />
37         </div>
38
39         <input id="add" type="button" data-icon="plus" value="Add a wish" style="border-radius:10px;background-color:#005aff;"/>
40       </form>
41
42     </div>
43
44
45
46 </div>
```

**Appendix 3: Code Listing Camera plugin camera.js**

```
2  * http://www.apache.org/licenses/LICENSE-2.0
3  */
4
5  var argscheck = require('cordova/argscheck'),
6      exec = require('cordova/exec'),
7      Camera = require('./Camera');
8
9  var cameraExport = {};
10
11 // Tack on the Camera Constants to the base camera plugin.
12 for (var key in Camera) {
13     cameraExport[key] = Camera[key];
14 }
15
16 cameraExport.getPicture = function(successCallback, errorCallback, options) {
17     argscheck.checkArgs('fF0', 'Camera.getPicture', arguments);
18     options = options || {};
19     var getValue = argscheck.getValue;
20
21     var quality = getValue(options.quality, 50);
22     var destinationType = getValue(options.destinationType, Camera.DestinationType.FILE_URI);
23     var sourceType = getValue(options.sourceType, Camera.PictureSourceType.CAMERA);
24     var targetWidth = getValue(options.targetWidth, -1);
25     var targetHeight = getValue(options.targetHeight, -1);
26     var encodingType = getValue(options.encodingType, Camera.EncodingType.JPEG);
27     var mediaType = getValue(options.mediaType, Camera.MediaType.PICTURE);
28     var allowEdit = !!options.allowEdit;
29     var correctOrientation = !!options.correctOrientation;
30     var saveToPhotoAlbum = !!options.saveToPhotoAlbum;
31     var popoverOptions = getValue(options.popoverOptions, null);
32     var cameraDirection = getValue(options.cameraDirection, Camera.Direction.BACK);
33
34     var args = [quality, destinationType, sourceType, targetWidth, targetHeight, encodingType,
35               mediaType, allowEdit, correctOrientation, saveToPhotoAlbum, popoverOptions, cameraDirection];
36
37     exec(successCallback, errorCallback, "Camera", "takePicture", args);
38     // XXX: commented out
39     //return new CameraPopoverHandle();
40 };
41
42
43 cameraExport.cleanup = function(successCallback, errorCallback) {
44     exec(successCallback, errorCallback, "Camera", "cleanup", []);
45 };
46
47 module.exports = cameraExport;
48
```