Design and Development of Personalized Contents-feed Apps Intended for Web-of-Things

Abhijit Suprem and Reza Raeisi

***Abstract*—This work focuses on design and development of a prototype App under Windows platform. The function of this App is that it dynamically consolidates News feeds for the intended users. The App can be configured to personalized mode as per users need. We have reviewed current research, and highlighted usage of available software packages and standards and their dependencies for app development process emphasizing both design and programming aspects. This paper also presents a holistic guideline on development of App type software which includes not only software flexibility, scalability, and configurability but also advanced design principles such as typography and aesthetics. The developed App is up and running under Windows 8 and is freely available to users.**

***Index Terms*— App design, Contents-feed, aggregation, Web-of-Things, flexibility, scalability, configurability, and modularity.**

I. INTRODUCTION

Due to the proliferation of mobile devices and tremendous advances in their physical and logical capabilities, end-user applications, commonly known as Apps, are becoming significantly important for everyday users of those devices such as smart phones, iPADs, tablets, and other portable devices [1]. Apps are the executable software packages uniquely designed for intended smaller applications for everyday users. There are various types of Apps available in the marketplace. The user can now monitor the current balance of his bank account figures at anytime and anywhere without opening the bank's website. All these became possible with the advent of App marketplace middleware model. For example, the Apple App Store introduced App ecosystem that is capable of providing interfaces and functional entities for the development of end-user Apps for any purposes [2].

Microsoft's Business and Marketing Strategy reported that there are now more than 70,000 Apps in the Windows Store and it is increasing steadily [3] (Fig.1). There are various categories of apps such as utility Apps, entertainment Apps, games Apps, travel App, social Apps, company Apps, and News Apps. Several other examples include Apps that facilitate record keeping, data logging, survey taking, and even time tracking. The Apps are being developed and becoming popular due to the reasons that (i) more and more people are converging their information retrieval strategies to contents-based setting so that on-the-spot and as-and-when access of information will be more feasible, and (ii) the Web-of-Things (WoT) [4] and Internet of Things (IoT) [5] have already drawn attention to an environment where the mobile and portable electronic devices, appliances and incredible numbers of Web and native Apps, commonly called Things, for optimum use by the everyday users. The embedded device need to be connected by fully integrating them to the World Wide Web (www). Thus, WoT and IoT envisioned that they would bring all physical and logical Things of the world into the network. The WoT and IoT are constantly evolving and have changed over to sprout new areas of growth [6].

Manuscript received July 16, 2013; revised August 1, 2013. Abhijit Suprem is with the Department of Electrical and Computer Engineering, California State University, Fresno, California, USA (e-mail: asuprem@mail.fresnostate.edu). Dr. Reza Raeisi is also with the Department of Electrical and Computer Engineering, California State University, Fresno, California, USA.

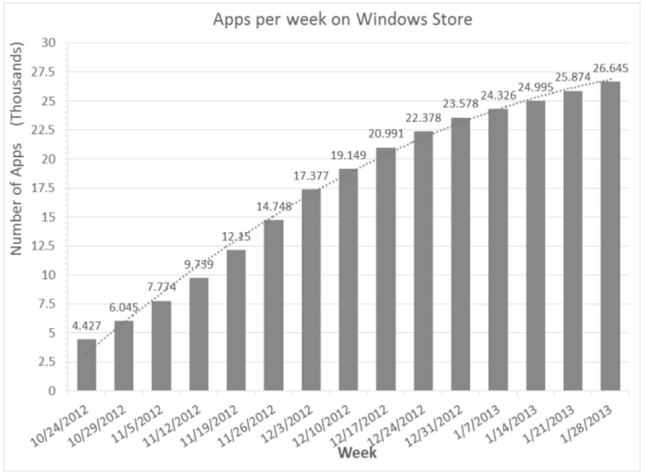


Fig.1. Increase in Apps (Data from techrepublic.com [7])

There has been a noticeable shift in software engineering and application development in the past two decades from single-user offline applications for large-scale tasks to mobile multi-platform, cloud-based Web-Apps. As mentioned, usage of mobile hardware such as smartphones and tablets has increased dramatically in the personal, public, and industrial workspace, prompting development of personalized and dedicated Apps to maximize efficiency and productivity. As a result, recently the annual business growth of mobile App development studios and companies is more than 100%

compound and remains at a high rate, unlike other businesses. For several years people have already become used to Web Apps within a web browser with rich Graphical User Interface (GUI) [8].

II. Review on Technology Platforms

Although the design and development of Apps inherit underlying similar methods and procedures as that of software development in a generalized sense, the version-based updating procedure is distinctly different [9]. This entails not only programmatic differences at the design level but also technical expertise on the interoperability and configurability while maintaining the Apps' sustainability. At the same time, the entire development process demands sound knowledge on the required part of the operating system (OS) such as application program interface (API), interoperable and configurable executives that support hardware and logical interfacing, programming tool/integrated development environment (IDE), middleware connectives, and the hardware platform including standards (commonly referred to as mobile platforms).

Currently, mobile platform architectures are built as extensions of the parent OS user experience (UX). For example, the three major competitors of mobile operating systems – iOS (Apple Inc.), Android (Google Inc.), and Windows RT (Microsoft Corp.) – retain major features and characteristics of their parent OS: Mac OSX, Google Web Services, and Windows NT, respectively. Consequently, each mobile OS provides specialized features to the developers and consumer. iOS offers a higher volume (currently at 1 million apps) of Apps for more diverse characteristic scenarios (e.g. tables, charts, and so on); its primary usage is consumer-centric rather than business-centric. Consequently, there are fewer numbers of business-based Apps on the iOS platform compared to other categories, such as entertainment, games, and lifestyle [10]. The Android platform straddles the fence between consumer-centric and business-centric profiles with a wider product and consumer base. Due to the open-source nature of the platform, there is a higher propagation of Android conformant tablets and smartphones in the market. While it opens up a larger market for developers, the inconsistencies in device specifications and a variety of device resolutions makes it difficult to develop apps that are compatible platform-wide [11]. The Windows 8 RT and its related Pro platform were released in late 2012, is currently the fastest growing App marketplace of the three mentioned [12] because it is the next iteration in the Windows NT development milestone from Windows 7, which has captured approximately 50% of portable electronics [12] (Fig. 2). The unique nature of Windows 8 as a desktop hybrid OS gives it a significantly larger market than its competitors and allows it to be used ubiquitously [13].

The App development process requires in depth knowledge of not only the API and its associated commands and syntax, but also holistic knowledge on aesthetic design guidelines for providing a better user experience. Currently, there are multiple publications that cover various aspects of App design such as (i) aesthetics, (ii) usability, (iii) typography, (iv) interactivity, and (v) visual scheme. There are also multiple publications covering software standards in Windows 8 app development. These publications offer lessons in programming in the language of their choices. However, they do not focus on the software engineering aspect of the app development, i.e. such publications, rather than focusing on the evolution of an App from prototype design to implementation to testing, instead focus on the language primarily as an encyclopedic guide [14-16]. The existing approaches make such guidelines useful as references, rather than actual guidelines on software development principles.

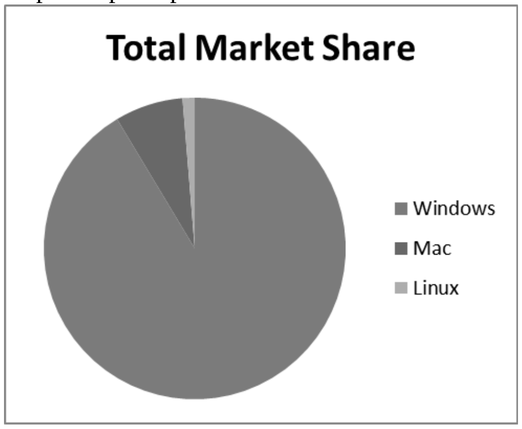


Fig.2. Market share of major OS

III. Motivation, Research Problem and objective

The motivation comes from the literature review, as mentioned above that the future is bright in regard to usage of Apps by each person holding a mobile device. As a result number of new Apps has significantly rose to high and every week the number is increasing. This means that there are now more than ever the numbers of App developers available in the World and their number is also increasing. However, the publications in regard to design and development and the software development design principles on Apps are scarce. We were then motivated to pursue research such that literature review, methodology, design principles, available standards and interfaces can be updated, and eventually present our work through a design of simple but representative App.

This work focuses on design and development of a prototype App under Windows platform. The function of this App is that it dynamically consolidates News feeds for the intended users. News Apps are applications that deliver news. The App can be configured to personalized mode as per users need. The App is flexible, configurable, and user-friendly.

IV. Methodology

App design is relatively less complex development process. The reason is that choosing the right prototyping tools for its development can be time consuming and their selection eventually lend to trial and errors. In order to selecting the prototyping tools, a reasonable catalog of tools can be built at the first place and be available. In order to achieve this, the first thing to do is create a classification of those tools. Once the tools are acquired, the specification phase starts.

C++ is the primary language for application development in the Windows. The advantages of C++ are its Simula-style programming, faster run-time execution, low overhead, support for multiple modeling and programming methods, such as procedural, object-oriented, and generic programming. C# is a C++ derivative that extensively utilizes the .NET framework, developed by Microsoft. It has various higher level language routines, as well as controlled access to hardware memory. The .NET framework is a software environment with various APIs and a large library of common tools and systems for developing applications in Windows OS. The C# language is closely tied to .NET libraries. XAML (Extensible Application Markup Language) is a close relative of .NET framework as it is a Windows OS-based markup language that maps to the .NET software environment (Common Language Runtime – CLR). While most XAML declarations can also be coded in C#, markup in XAML has less complexity and is easier to process.

Javascript is a high-level object oriented web-based scripting language used in conjunction with HTML (HyperText Markup Language). Due to Windows 8 support for Apps written in HTML/Javascript, Javascript, previously consigned to Web Apps, is now an important component of Windows OS programming.

The App was developed under the Windows 8 RT platform utilizing HTML5 and CSS3 (Cascading Style Sheets) standards, Windows Library for JavaScript, and in all the above mentioned tools and ingredients in the Visual Studio 2012 IDE. A laptop PC was used for the development of App.

V. Design Approaches and Principles

Above all, the paper considers underlying knowledgebase required for the design of App. Bearing in mind that current publications regarding software engineering guidelines for App development on Windows 8 RT platform emphasize either the design or the programming aspect, this paper also presents holistic guidelines on development of App type packaged software. Thus, one of the objectives of this paper is to present the involved design process in App development. Although, there are documentations that cover rules on how to use C#, C++, .NET, XAML, and JavaScript APIs, there is a dearth of guidelines covering structural and developmental phases [9] of App design taking into account of not only software flexibility, scalability, and configurability but also advanced design principles such as aesthetics and typography.

Performance of the design processes can be improved by knowing such parameters as time of adoption, number of users, marketing, and development phase. The process also involves hardware and software types and compatibility, the nature of web browsing and available standards (whether proprietary or open system). Much concentration is given on management layer and physical layer of the layered model as schematically shown in Fig.3.

The use of second software at the top of the conventional system level software play vital role in development process. This level of software sometimes referred as simulation tool, but in case of App design, it is simply a modeling environment. The applicability of modeling environment is considered as an additional requirement that can significantly improve the performance of the design processes. The modeling phase would not only reduce the design time compared to conventional method but also validate the entire design. There are several tools available to model the Apps: UML (Unified Modeling Language) works fine. While developing the models, the relationship (internal features of the software objects) between the model elements can be established and verified. The simulation tools facilitate management of App features and functionality [17]. The model accommodates the services for achieving internal communication between functional entities as well as management functions for the overall App for sustainability.

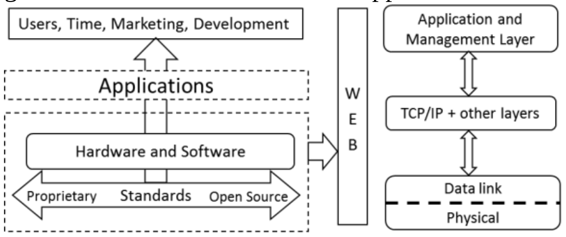


Fig.3. Design requirement

Overall, the development includes four stages: App specification, coding and integration, relationships specification, and testing and validation as illustrated in Fig.4. Prior to the development process, it is necessary to understand the user needs and, by extension, product requirements.

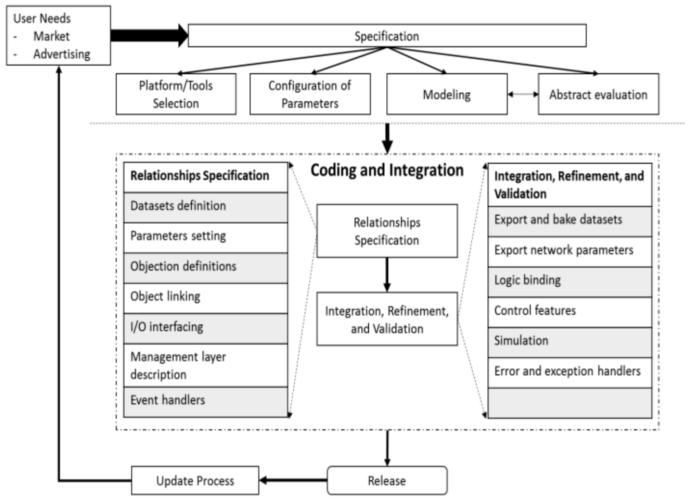


Fig.4. Design principle

A. Specification

The specification stage involves building a conceptual framework of the App based on user needs.

1) Platform/tools selection

The proper OS platform and the programming tool/IDE must be chosen for the App development. Each of the platforms (iOS, Android, Windows RT) provides a specialized IDE for App creation within that platform.

2) Configuration of parameters

Parameters for the App are initially defined in this section for use as a reference during the coding and integration section.

3) Modeling

A basic, limited user interface (UI) is designed for the App. This forms the basis of the future design aspect as a guide towards the finalized design.

4) Abstract evaluation

Once the blue print form of the design is over, offline evaluation is needed beforehand. A spreadsheet may be needed to record for possible future addition or a link to another App.

B. Coding and Integration

This stage entails the construction and debug deployments of the App's resources and binaries in an app package. Each of the sub-stages includes components such as integration, relationships, validation, and refinement integral to the entire App.

1) Relationships specifications:

The App requires various parameters, external sources of data, internal datasets, and event handlers to add runtime functionality. Each component contributes additional feature sets to the App.

a) Datasets definition: External and internal data sources such as settings, user accounts, and user data (placeholders are included until the end users enter their data) are defined and connected to the App.

b) Parameters setting: Parameters integral to the App's functionality, e.g. number of elements in a Periodic Table App, or number of sources to retrieve in a content aggregation App, are included.

c) Object definitions: Since all the major App development IDEs provide object oriented languages only, runtime objects are integral to the App. Objects must be predefined before they are called.

d) I/O interfacing: Hardware interfacing, if necessary, is defined in this sub-stage. Necessary APIs (e.g. webcam API, microphone API, etc.) can also be linked.

e) Management layer description: The management layer consists of higher level objects that control the App's core functionality. These entities ensure the App's scalability by accommodating larger datasets and event handlers.

f) Event handlers: The response time to various events are collated and defined. Each event handler is executed asynchronously with the primary function.

2) Integration, refinement, and validation

Upon specifications of resources, the App's coding and design are tested, refined, and validated with the following methods.

a) Export and bake datasets: Datasets are exported from editable format and baked. The latter process formats the datasets in order to enable rapid access of data during runtime. This is especially important in larger databases.

b) Export network parameters: The App's network parameters such as online synchronization and external data source addresses are also exported and baked to facilitate faster data retrieval.

c) Logic binding: The App's code component must perform flawlessly at runtime. Errors that occur should have handlers to prevent crashing or unwanted interrupts.

d) Control features: Several features that require user control are tested and the necessary protocols implemented to give the App a better UX in terms of user choice.

e) Simulation: During App simulation, scenarios are implemented to test the App's flexibility, scalability, configurability, and interoperability in its executable state. The simulations can reveal sections that require refinement and aids in validation.

f) Error and exception handlers: Some errors and exceptions detected during simulation, such as failure to access network due to disabled Wi-Fi or absence of data due to missing components require handlers to prevent App crashes and to provide users information on solving these problems. In the case of the former, it is required to turn on Wi-Fi.

Further, the design should have certain characteristics in order to make it both user-friendly (navigable and interactive) and developer-friendly (flexibility, scalability, and configurability) – necessary components of app design [18]. We have adopted the design principles that are presented in [19-20].

VI. Design Features and Attributes

This research is unique because it approaches the research problem from a combined software engineering and design perspective, compared with existing design-only or language-only presentation and publications. A two-tier hierarchy is utilized to embed design requirements (Fig 5). Fig.6 shows the software components.

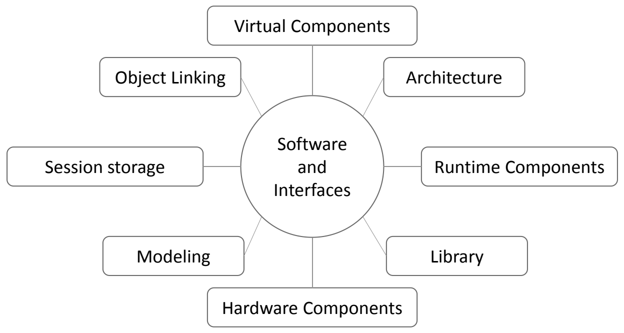


Fig.5. Software and interfaces

In the developer tier, flexibility is essential in software engineering. The app used dynamic data storage over the cloud in order to preserve layout across multiple sessions. The app utilized the Windows JavaScript Library to implement cloud backup. Flexibility is best implemented in HTML5 using CSS3 templates imported into multiple pages.

Scalability involves an app retaining core user experience regardless of expansion. This involves another attribute – modularity, which can be implemented by storing non-core pages, such as posts for a social client, lessons for an educational client, or levels for a game app, not as app data, but as an external dataset than can be read and implemented. While this approach is most widely used in social clients, other app types often directly include non-core content, making the app less scalable due to sluggishness from larger host system memory footprint. Configurability is implemented by utilizing JavaScript to modify static markup based on user input. Rather than using a single color schema, for example, configurability can be introduced to change the color schema with user choice. This change can be implemented by storing the setting in an external file or in the cloud. Each instance of the app can retrieve the stored data dynamically update the CSS3 style sheet with JavaScript. With regard to aesthetics, the Microsoft publications cover the various aspects of design. The research utilizes existing publications and incorporates the design principles into the software engineering principles developed in this guideline [21]. These three attributes in the developer tier are the core software engineering principles behind app development.

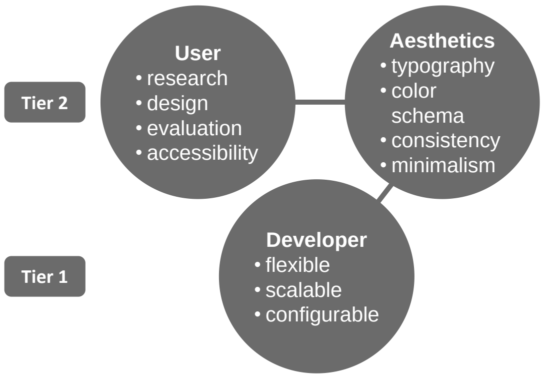
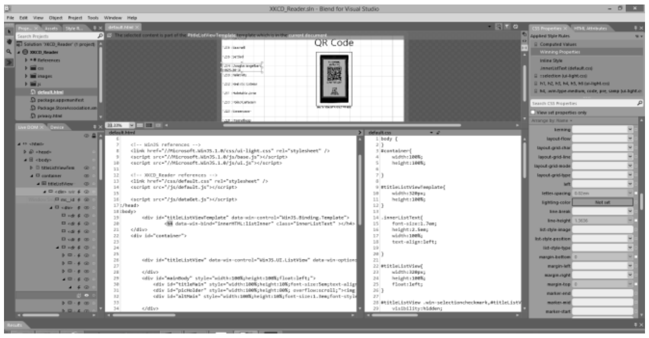


Fig.6. Two-tiered hierarchy

VII. Results and Contributions

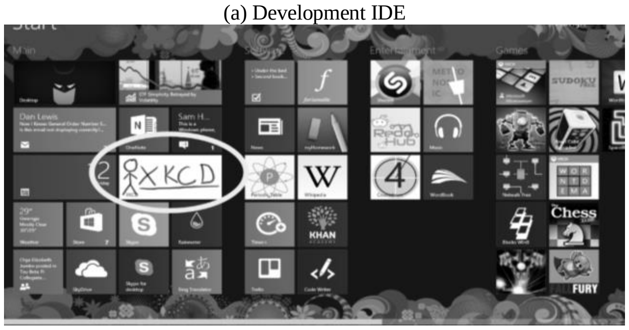
A guideline for the development of App was outlined based on the established software development principles. The availability of guideline is considered as a resource for design and development approaches. Utilizing the guidelines, a prototype App was designed for the Windows 8 RT platform. First, we prepared a table taking into account of standards, language, GUI, API, and connectivity. Within each domain technical questions are set and answered. The parameters cited under Specification and Coding and Integration presented in section V were considered in developing the proposed App. Further, aspects of software engineering such as operational flexibility, scalable design, and configurable front-end were addressed. The prototype App called "ReadXKCD". The second representative App called "Reader by Zen" is currently in post-production stage. The snapshots of both the Apps are shown in Fig.7. The former one is an App that retrieves publications from web comic sites XKCD.com. If the user has multiple devices (WoT) and has signed-up, it syncs with all the devices and it also automatically maintain records of the readings already completed. The second one is an App that aggregates thematic news (economics, technology, politics, movie, etc.) from multiple sources on the same theme. That is the App is a dynamic news aggregator that retrieves relevant content from various websites. The data is organized categorically and displayed to the user. Each category is topical. In addition, there is a "Front Page" category, which is an amalgamation of topical content most accessed by the user. The "Front Page" is dynamic and reflects the user's interest in the content. The user's focus on one category of news is reflected in that same category's increase visibility in the "Front Page." In conjunction to the "Front Page", the app continuously retrieves content from the websites and updates the display.



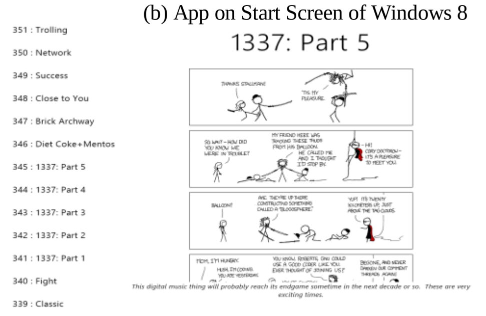
```
<!-- WinJS references -->
<link href="//Microsoft.WinJS.1.0/css/ui-light.css" rel="stylesheet" />
<script src="//Microsoft.WinJS.1.0/js/base.js"></script>
<script src="//Microsoft.WinJS.1.0/js/ui.js"></script>

<!-- XKCD_Reader references -->
<link href="/css/default.css" rel="stylesheet" />
<script src="/js/default.js"></script>

<script src="/js/dataGet.js"></script>
</head>
<body>
    <div id="titleListViewTemplate" data-win-control="WinJS.Binding.Template">
        <h4 data-win-bind="innerHTML:listInner" class="innerListText" ></h4>
    </div>
    <div id="container">


    <div id="titleListView" data-win-control="WinJS.UI.ListView" data-win-options

    </div>
    <div id="mainBody" style="width:100%;height:100%;float:left;">
        <div id="titleMain" style="width:100%;height:10%;font-size:5em;text-alig
        <div id="picHolder" style="width:100%;height:80%; overflow:scroll;"><img
        <div id="altMain" style="width:100%;height:10%;font-size:1.3em;font-style

    </div>
```

```
body {
}
#container{
    width:100%;
    height:100%;

}

#titleListViewTemplate{
    width:320px;
    height:100%;
}

.innerListText{
    font-size:1.7em;
    height:2.5em;
    width:100%;
    text-align:left;

}

#titleListView{
    width:320px;
    height:100%;
    float:left;
}

#titleListView .win-selectioncheckmark,#titleListV
    visibility:hidden;
```

(a) Development IDE



(b) App on Start Screen of Windows 8



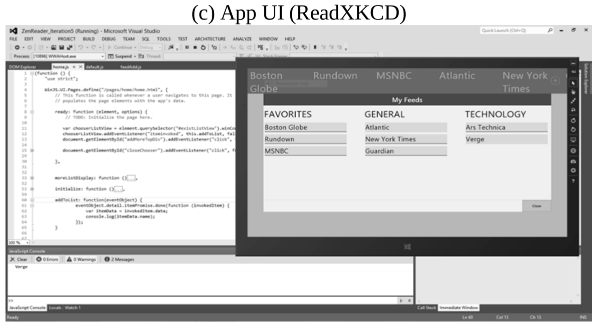
351 : Trolling
350 : Network
349 : Success
348 : Close to You
347 : Brick Archway
346 : Diet Coke+Mentos
345 : 1337: Part 5
344 : 1337: Part 4
343 : 1337: Part 3
342 : 1337: Part 2
341 : 1337: Part 1
340 : Fight
339 : Classic

1337: Part 5

THANKS STALLMAN!
'TIS MY PLEASURE.
SO WAIT—HOW DID YOU KNOW WE WERE IN TROUBLE?
MY FRIEND HERE WAS TRACKING THESE THUGS FROM HIS BALLOON.
HE CALLED ME AND I THOUGHT I'D STOP BY.
—HI! CORY DOCTOROW—IT'S A PLEASURE TO MEET YOU.
BALLOON?
AYE. THEY'RE UP THERE CONSTRUCTING SOMETHING CALLED A "BLOGOSPHERE."
YUP! IT'S TWENTY KILOMETERS UP, JUST ABOVE THE TAG CLOUDS.
MOM, I'M HUNGRY.
HUSH, I'M COOKING. YOU ATE YESTERDAY.
YOU KNOW, ROBERTS, YOU COULD USE A GOOD CODER LIKE YOU. EVER THOUGHT OF JOINING US?
BEGONE, AND NEVER DARKEN OUR COMMENT THREADS AGAIN!

This digital music thing will probably reach its endgame sometime in the next decade or so. These are very exciting times.

(c) App UI (ReadXKCD)


ZenReader_Iteration5 (Running) - Microsoft Visual Studio
FILE EDIT VIEW PROJECT BUILD DEBUG TEAM SQL TOOLS TEST ARCHITECTURE ANALYZE WINDOW HELP
Process: [10896] WWAHost.exe
Suspend
DOM Explorer
home.js
default.js
feedAdd.js
JavaScript Console
Clear
0 Errors
0 Warnings
2 Messages
Locals
Watch 1
Call Stack
Immediate Window
Ready

```
(function () {
    "use strict";

    WinJS.UI.Pages.define("/pages/home/home.html", {
        // This function is called whenever a user navigates to this page. It
        // populates the page elements with the app's data.

        ready: function (element, options) {
            // TODO: Initialize the page here.

            var chooserListView = element.querySelector("#existListView").winCo
            chooserListView.addEventListener("iteminvoked", this.addToList, fal
            document.getElementById("addMoreTopDiv").addEventListener("click",

            document.getElementById("closeChooser").addEventListener("click", f

        },


        moreListDisplay: function () {...},

        initialize: function () {...},

        addToList: function(eventObject) {
                eventObject.detail.itemPromise.done(function (invokedItem) {
                    var itemData = invokedItem.data;
                    console.log(itemData.name);
                });
        }
```

Verge

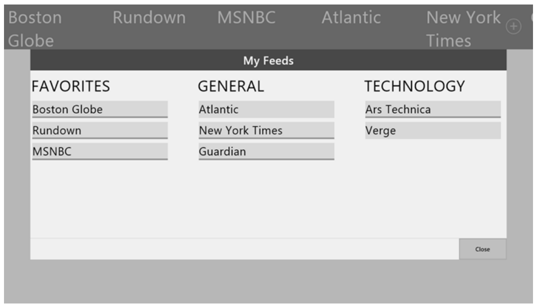
Boston Globe Rundown MSNBC Atlantic New York Times

My Feeds

FAVORITES	GENERAL	TECHNOLOGY
Boston Globe	Atlantic	Ars Technica
Rundown	New York Times	Verge
MSNBC	Guardian	

Close

(d) App UI (Reader by Zen) during debug



(e) App UI (Reader by Zen) during testing

Fig.7. Snapshots of developed Apps (a-c): ReadXKCD, (d-e): Reader by Zen.

In both the cases, the Apps feed contents of the article based on the users' choice, thus personalized. The retrieval and display processes are asynchronous, reducing network load and improving processor footprint. Due to its modular nature, each feature of the app is independent of other features. Further, modifications to a module do not require significant changes to peripheral modules, reducing size of updates and improving workflow. The app is also scalable. For example, the "Reader by Zen" can be used for small-scale aggregation of topical news, as well as large-scale aggregation of multiple sources and categories. Regardless of size of the input data, the App dynamically updates both the categorical content and the "Front Page."

VIII. Conclusions

In this paper, we demonstrated design and development of a prototype App under Windows platform. The developed App dynamically consolidates News feeds for the users. The App is configurable as per users' needs. Current guidelines for app development emphasize either the design or the programming aspect without much focus on the other. The integration of programming and designs lends itself when significant changes or updates are to be made. This paper also presents a guideline on development of App type software. An App that closely follows the guideline has a better workflow as all programmatic and design elements are inherently tied together. The guideline focuses not only on software flexibility, scalability, and configurability but also advanced design principles such as typography and aesthetics. Therefore, necessary changes to one aspect must also be replication on the other, preserving the cohesive characteristic of the App. The App uses tools and systems from Microsoft Corp. and follows the Web-of-Things concept in the design and development. Further, a guideline on multiplatform App design and development was created to address the twin issues of software engineering and visual design principles.

Acknowledgment

I would like to acknowledge Dr. Honora Chapman, Director, Smittcamp Family Honors College, for her advice and encouragement. I would like to acknowledge Dr. Reza Raeisi for giving me the opportunity to further develop the guideline. I would further like to acknowledge Dr. Nagy Bengiamin, Chair of Electrical and Computer Engineering and Dr. Ram Nunna, Dean of Lyles College of Engineering for providing the computing facilities for this research.

References

[1] Kimbler, Kristofer (2010). App store strategies for service providers, 2010 14th International Conference on Intelligence in Next Generation Networks (ICIN), pp 1-5, Oct. 11-14.

[2] Zhung-Xun Liao, Po-Ruey Lei, Tsu-Jou Shen, Shou-Chung Li, Wen-Chih Peng (2012). AppNow: Predicting Usages of Mobile Applications on Smart Phones, Conference on Technologies and Applications of Artificial Intelligence (TAAI), pp.300-303, Nov. 16-18

[3] John Callaham (2013). Report: Windows Store reaches 80,000 Windows 8 app mark in June, Neowin.net.

http://www.neowin.net/news/report-windows-store-reaches-80000-windows-8-app-mark-in-june

[4] Yu Chenzhou, Wei Jia Shen, Wen Sun, Gang Hu, Jing Li, Lin Huao Xu, Xin Zhi Sun, Yue Pan (2012). Active Linked Data for Human Centric Semantic Web of Things, *IEEE International Conference on Green Computing and Communications*, vol.9, no. 16, pp. 20-23 doi: 10.1109/GreenCom.2012.12

[5] L. Coetzee, J. Eksteen (2011). The Internet of Things - promise for the future? An introduction, *IST-Africa Conference Proceedings*, vol., no., pp.1,9, pp. 11-13

[6] S. S. Mathew, Y. Atif, Q. Z. Sheng, Z. Maamar (2011). Web of Things: Description, Discovery and Integration, *International Conference on and 4th International Conference on Cyber, Physical and Social Computing*, vol. 9, no. 15, pp. 19-22. doi: 10.1109/iThings/CPSCom.2011.165

[7] Shultz, Greg. "The Countdown to 100,000 Windows 8 Apps Falls Short." TechRepublic. TechRepublic, 30 Jan. 2013. Web. 20 Apr. 2013. <http://www.techrepublic.com/blog/windows-and-office/the-countdown-to-100000-windows-8-apps-falls-short/?s_cid=e064>.

[8] H. Samir and A. Kamel (2007). Automated reverse engineering of Java graphical user interfaces for web migration, *5th International Conference on Information and Communications Technology, ICICT*. vol. 157, no. 162, pp. 16-18. doi: 10.1109/ITICT.2007.4475638

[9] Abhijit Suprem and Gregory Kriehn (2012). Effective Web Design for Content Managers and End-Users, *33rd Annual Central California Research Symposium*, California State University, Fresno, USA.

[10] Hwu, C., Kirchhoff, E. (2012). Sensational iOS Design: First Principles and New Trends for 2012. *iOS Developer Meetup*.

[11] J. Pappano (2011). Android Fragmentation: Solving the Enterprise Mobility Dilemma, *MaaS360 Fiberlink Communications*

[12] NetMarketShare Reports on Operating System Market Share (by Version). http://www.netmarketshare.com/

[13] Wheeler David (2012). *The Metro Design Language*. Presented at DevWeek 2012, London, England, UK

[14] Brockschmidt, Kraig. *Programming Windows 8 Apps with HTML, CSS, and Javascript*.[S.l.]: Microsoft, 2012. Print.

[15] Mark Pilgrim (2010). *HTML5: Up and Running*. O'Reily Media. 2010

[16] Chris Sells (2012). *Building Windows 8 Apps with JavaScript (Early Edition)*. Addision-Wesley Professional. 2012. Print

[17] N. P. Mahalik and Kiseon Kim (2008). A prototype for hardware-in-the-loop simulation of distributed control architecture, *IEEE Transaction on Systems, Man and Cybernetics (Part C), Vol. 38, Iss. 2*, pp. 189-200.

[18] Andy Rich (2013). Introducing C++/CX and XAML. *Windows 8 Special Issue – MSDN Magazine*, http://msdn.microsoft.com/en-us/magazine/jj651573.aspx

[19] Schooley, Brent. (2012). F*illing in the UX gaps in Metro Style Apps*. Presented at TechBash 2012, Luzerne, Pennsylvania

[20] S. Mishra and N. P. Mahalik (2011). Virtual distributed control systems (DCS) and specification, IEEE Int'l conf. on Process Automation, Control and Computing, PACC 2011, July 2011, Coimbatore, India.

[21] Guidelines for Fonts (Windows Store Apps) *Microsoft Developer Network (MSDN Magazine)* (2012). http://msdn.microsoft.com/en-us/library/windows/apps/hh700394.aspx