

KIEL UNIVERSITY OF APPLIED SCIENCES

MASTER THESIS

---

# Developing Cross-Platform Mobile Application Solution Using Xamarin and Microsoft Azure

---

*Author:*  
Ranjith MURTHY

*Supervisor:*  
Dr. Jens LÜSSEM

*A thesis submitted in fulfillment of the requirements  
for the degree of Master of Science in Information Technology*

*in*

Department of Computer Science and Electrical Engineering  
Kiel University of Applied Sciences



April 5, 2017



## Declaration of Authorship

I, Ranjith MURTHY, declare that this thesis titled, “Developing Cross-Platform Mobile Application Solution Using Xamarin and Microsoft Azure” and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed:

---

Date:

---



*“Thanks to my solid academic training, today I can write hundreds of words on virtually any topic without possessing a shred of information, which is how I got a good job in journalism.”*

Dave Barry



## *Abstract*

The global mobile market has experienced a tremendous increase in the number of smart computing device users per household. The mobile device base is strongly divided between different mobile platforms, most importantly Android, iOS and Windows Phone. Applications developed for one platform with traditional development methods only work on that platform, and supporting multiple platforms requires developing the application separately for each of the platforms.

This trend affects small business industry whose user spread across different mobile operating systems. Such companies strive to decrease the development time and with a satisfying solution that is delivered on time. To decrease the time-to-market and thus the cost of the final product, companies seek to develop applications independent of the target mobile operating system by using a cross-platform approach. This approach can eliminate the increased effort that normally comes with developing a separate application for each mobile operating system, providing a more efficient solution.

The aim of this thesis describes How to develop a cost effective multi-target mobile solution strategy for “Customer Feedback Management System for Restaurants”. The application targets the leading mobile operating systems iOS Android and Windows. While sharing the business logic through a portable class library. The Xamarin and Microsoft Azure toolchain provides the mechanics for compiling the C# codebase to run on across different Operating platform.

During the research phase focus on finding the suitable data analysis method to analyses the user feedback data. In addition, finding suitable visualization techniques to visualize analyzed feedback data.





## *Acknowledgements*

I would first like to thank my thesis supervisor Dr. Jens LÜSSEM of the Department of Computer Science and Electrical Engineering at Kiel University of Applied Sciences. The door to Prof. LÜSSEM's office was always open whenever I ran into a trouble spot or had a question about my research or writing. He consistently allowed this paper to be my own work, but steered me in the right direction whenever he thought I needed it.

I would also like to acknowledge Dr. Stephan Schneider of the Department of Computer Science and Electrical Engineering at Kiel University of Applied Sciences as the second supervisor of this thesis, and I am gratefully indebted to him for his very valuable comments on this thesis.

Finally, I must express my very profound gratitude to my parents and to my brother for providing me with unfailing support and continuous encouragement throughout my years of study and through the process of researching and writing this thesis. This accomplishment would not have been possible without them.

Thank you.

Author

Ranjith MURTHY



# Contents

<b>Declaration of Authorship</b>	<b>iii</b>
<b>Abstract</b>	<b>vii</b>
<b>Acknowledgements</b>	<b>ix</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Background . . . . .	1
1.2 Problem Statement and Research Questions . . . . .	1
1.2.1 A (not so short) Introduction to L <sup>A</sup> T <sub>E</sub> X . . . . .	1
1.3 Structure of the Thesis . . . . .	2
<b>2 Methodology</b>	<b>3</b>
2.1 Literature Review . . . . .	3
2.2 Learning L <sup>A</sup> T <sub>E</sub> X . . . . .	3
2.2.1 A (not so short) Introduction to L <sup>A</sup> T <sub>E</sub> X . . . . .	3
2.2.2 A Short Math Guide for L <sup>A</sup> T <sub>E</sub> X . . . . .	4
2.2.3 Common L <sup>A</sup> T <sub>E</sub> X Math Symbols . . . . .	4
2.2.4 L <sup>A</sup> T <sub>E</sub> X on a Mac . . . . .	4
2.3 Getting Started with this Template . . . . .	4
<b>3 Mobile Application Development</b>	<b>5</b>
3.1 Mobile Platforms . . . . .	5
3.1.1 Android . . . . .	5
3.1.2 iOS . . . . .	6
3.1.3 Windows . . . . .	7
3.1.4 Tizen . . . . .	8
3.2 Mobile Applications . . . . .	9
3.2.1 Native Applications . . . . .	9
3.2.2 Mobile Web Apps . . . . .	10
3.2.3 Hybrid Mobile Applications . . . . .	10
3.2.4 Conclusion . . . . .	11
3.3 Cross-Platform Development Tools . . . . .	11
3.3.1 PhoneGap . . . . .	11
3.3.2 Titanium . . . . .	12
3.3.3 Xamarin . . . . .	14
<b>4 Introduction to Cross-platform Development with Xamarin</b>	<b>15</b>
4.1 Understanding the Xamarin Mobile Platform . . . . .	15
4.2 Architecture . . . . .	15
4.3 Setting Up A Xamarin Cross Platform Solution . . . . .	15
4.3.1 Code Sharing Options . . . . .	16
4.3.2 Portable Class Libraries . . . . .	16

4.3.3	Shared Projects . . . . .	16
4.3.4	.NET Standard . . . . .	16
4.4	Dealing with Multiple Platforms . . . . .	16
4.5	Practical Code Sharing Strategies . . . . .	17
4.6	Dealing with Multiple Platforms . . . . .	17
4.7	Cross-Platform User Interfaces with Xamarin.Forms . . . . .	17
4.7.1	eXtensible Application Markup Language (XAML) . . . . .	18
4.8	Testing and App Store Approvals . . . . .	18
4.8.1	About this Template . . . . .	18
<b>5</b>	<b>Introduction to Microsoft azure</b>	<b>19</b>
5.1	Connected Services in Xamarin Studio . . . . .	19
5.2	Azure App Services . . . . .	19
5.2.1	A (not so short) Introduction to L <sup>A</sup> T <sub>E</sub> X . . . . .	19
5.2.2	A Short Math Guide for L <sup>A</sup> T <sub>E</sub> X . . . . .	20
5.2.3	Common L <sup>A</sup> T <sub>E</sub> X Math Symbols . . . . .	20
5.2.4	L <sup>A</sup> T <sub>E</sub> X on a Mac . . . . .	20
5.3	Active Directory Authentication . . . . .	20
5.3.1	WebAPI . . . . .	20
<b>6</b>	<b>Chapter Title Here</b>	<b>23</b>
6.1	Welcome and Thank You . . . . .	23
6.2	Learning L <sup>A</sup> T <sub>E</sub> X . . . . .	23
6.2.1	A (not so short) Introduction to L <sup>A</sup> T <sub>E</sub> X . . . . .	23
<b>7</b>	<b>Chapter Title Here</b>	<b>25</b>
7.1	Welcome and Thank You . . . . .	25
7.2	Learning L <sup>A</sup> T <sub>E</sub> X . . . . .	25
7.2.1	A (not so short) Introduction to L <sup>A</sup> T <sub>E</sub> X . . . . .	25
<b>8</b>	<b>Chapter Title Here</b>	<b>27</b>
8.1	Welcome and Thank You . . . . .	27
8.2	Learning L <sup>A</sup> T <sub>E</sub> X . . . . .	27
8.2.1	A (not so short) Introduction to L <sup>A</sup> T <sub>E</sub> X . . . . .	27
8.2.2	A Short Math Guide for L <sup>A</sup> T <sub>E</sub> X . . . . .	28
8.2.3	Common L <sup>A</sup> T <sub>E</sub> X Math Symbols . . . . .	28
8.2.4	L <sup>A</sup> T <sub>E</sub> X on a Mac . . . . .	28
<b>9</b>	<b>Chapter Title Here</b>	<b>29</b>
9.1	Welcome and Thank You . . . . .	29
9.2	Learning L <sup>A</sup> T <sub>E</sub> X . . . . .	29
9.2.1	A (not so short) Introduction to L <sup>A</sup> T <sub>E</sub> X . . . . .	29
<b>10</b>	<b>Conclusion</b>	<b>31</b>
10.1	Goal Fulfilment . . . . .	31
10.2	Future Work . . . . .	31
<b>A</b>	<b>Frequently Asked Questions</b>	<b>33</b>
A.1	How do I change the colors of links? . . . . .	33
	<b>Bibliography</b>	<b>35</b>

# List of Figures



# List of Tables

3.1	Smartphone Operating System market shares from years 2015 to 2016.	5
3.2	Native App Development across different platform . . . . .	10
3.3	The Mobile App Comparison Chart: Native vs. Mobile Web vs. Hybrid . . . . .	12





# List of Abbreviations

**LAH** List Abbreviations **Here**  
**WSF** What (it) Stands For



# Physical Constants

Speed of Light  $c_0 = 2.997\,924\,58 \times 10^8 \text{ m s}^{-1}$  (exact)



# List of Symbols

$a$	distance	m
$P$	power	W (J s <sup>-1</sup> )
$\omega$	angular frequency	rad



*For/Dedicated to/To my...*





## Chapter 1

# Introduction

### 1.1 Background

Welcome to this L<sup>A</sup>T<sub>E</sub>X Thesis Template, a beautiful and easy to use template for writing a thesis using the L<sup>A</sup>T<sub>E</sub>X typesetting system.

If you are writing a thesis (or will be in the future) and its subject is technical or mathematical (though it doesn't have to be), then creating it in L<sup>A</sup>T<sub>E</sub>X is highly recommended as a way to make sure you can just get down to the essential writing without having to worry over formatting or wasting time arguing with your word processor.

L<sup>A</sup>T<sub>E</sub>X is easily able to professionally typeset documents that run to hundreds or thousands of pages long. With simple mark-up commands, it automatically sets out the table of contents, margins, page headers and footers and keeps the formatting consistent and beautiful. One of its main strengths is the way it can easily typeset mathematics, even *heavy* mathematics. Even if those equations are the most horribly twisted and most difficult mathematical problems that can only be solved on a super-computer, you can at least count on L<sup>A</sup>T<sub>E</sub>X to make them look stunning.

### 1.2 Problem Statement and Research Questions

L<sup>A</sup>T<sub>E</sub>X is not a WYSIWYG (What You See is What You Get) program, unlike word processors such as Microsoft Word or Apple's Pages. Instead, a document written for L<sup>A</sup>T<sub>E</sub>X is actually a simple, plain text file that contains *no formatting*. You tell L<sup>A</sup>T<sub>E</sub>X how you want the formatting in the finished document by writing in simple commands amongst the text, for example, if I want to use *italic text for emphasis*, I write the `\emph{text}` command and put the text I want in italics in between the curly braces. This means that L<sup>A</sup>T<sub>E</sub>X is a "mark-up" language, very much like HTML.

#### 1.2.1 A (not so short) Introduction to L<sup>A</sup>T<sub>E</sub>X

If you are new to L<sup>A</sup>T<sub>E</sub>X, there is a very good eBook – freely available online as a PDF file – called, "The Not So Short Introduction to L<sup>A</sup>T<sub>E</sub>X". The book's title is typically shortened to just *lshort*. You can download the latest version (as it is occasionally updated) from here: <http://www.ctan.org/tex-archive/info/lshort/english/lshort.pdf>

It is also available in several other languages. Find yours from the list on this page: <http://www.ctan.org/tex-archive/info/lshort/>

It is recommended to take a little time out to learn how to use L<sup>A</sup>T<sub>E</sub>X by creating several, small 'test' documents, or having a close look at several templates on:

<http://www.LaTeXTemplates.com>

Making the effort now means you're not stuck learning the system when what you *really* need to be doing is writing your thesis.

### 1.3 Structure of the Thesis

If you are familiar with  $\text{\LaTeX}$ , then you should explore the directory structure of the template and then proceed to place your own information into the *THESIS INFORMATION* block of the **main.tex** file. You can then modify the rest of this file to your unique specifications based on your degree/university. Section ?? on page ?? will help you do this. Make sure you also read section ?? about thesis conventions to get the most out of this template.

If you are new to  $\text{\LaTeX}$  it is recommended that you carry on reading through the rest of the information in this document.

Before you begin using this template you should ensure that its style complies with the thesis style guidelines imposed by your institution. In most cases this template style and layout will be suitable. If it is not, it may only require a small change to bring the template in line with your institution's recommendations. These modifications will need to be done on the **MastersDoctoralThesis.cls** file.

## Chapter 2

# Methodology

## 2.1 Literature Review

Welcome to this L<sup>A</sup>T<sub>E</sub>X Thesis Template, a beautiful and easy to use template for writing a thesis using the L<sup>A</sup>T<sub>E</sub>X typesetting system.

If you are writing a thesis (or will be in the future) and its subject is technical or mathematical (though it doesn't have to be), then creating it in L<sup>A</sup>T<sub>E</sub>X is highly recommended as a way to make sure you can just get down to the essential writing without having to worry over formatting or wasting time arguing with your word processor.

L<sup>A</sup>T<sub>E</sub>X is easily able to professionally typeset documents that run to hundreds or thousands of pages long. With simple mark-up commands, it automatically sets out the table of contents, margins, page headers and footers and keeps the formatting consistent and beautiful. One of its main strengths is the way it can easily typeset mathematics, even *heavy* mathematics. Even if those equations are the most horribly twisted and most difficult mathematical problems that can only be solved on a super-computer, you can at least count on L<sup>A</sup>T<sub>E</sub>X to make them look stunning.

## 2.2 Learning L<sup>A</sup>T<sub>E</sub>X

L<sup>A</sup>T<sub>E</sub>X is not a WYSIWYG (What You See is What You Get) program, unlike word processors such as Microsoft Word or Apple's Pages. Instead, a document written for L<sup>A</sup>T<sub>E</sub>X is actually a simple, plain text file that contains *no formatting*. You tell L<sup>A</sup>T<sub>E</sub>X how you want the formatting in the finished document by writing in simple commands amongst the text, for example, if I want to use *italic text for emphasis*, I write the `\emph{text}` command and put the text I want in italics in between the curly braces. This means that L<sup>A</sup>T<sub>E</sub>X is a "mark-up" language, very much like HTML.

### 2.2.1 A (not so short) Introduction to L<sup>A</sup>T<sub>E</sub>X

If you are new to L<sup>A</sup>T<sub>E</sub>X, there is a very good eBook – freely available online as a PDF file – called, "The Not So Short Introduction to L<sup>A</sup>T<sub>E</sub>X". The book's title is typically shortened to just *lshort*. You can download the latest version (as it is occasionally updated) from here: <http://www.ctan.org/tex-archive/info/lshort/english/lshort.pdf>

It is also available in several other languages. Find yours from the list on this page: <http://www.ctan.org/tex-archive/info/lshort/>

It is recommended to take a little time out to learn how to use L<sup>A</sup>T<sub>E</sub>X by creating several, small 'test' documents, or having a close look at several templates on:

<http://www.LaTeXTemplates.com>

Making the effort now means you're not stuck learning the system when what you *really* need to be doing is writing your thesis.

### 2.2.2 A Short Math Guide for L<sup>A</sup>T<sub>E</sub>X

If you are writing a technical or mathematical thesis, then you may want to read the document by the AMS (American Mathematical Society) called, "A Short Math Guide for L<sup>A</sup>T<sub>E</sub>X". It can be found online here: <http://www.ams.org/tex/amslatex.html> under the "Additional Documentation" section towards the bottom of the page.

### 2.2.3 Common L<sup>A</sup>T<sub>E</sub>X Math Symbols

There are a multitude of mathematical symbols available for L<sup>A</sup>T<sub>E</sub>X and it would take a great effort to learn the commands for them all. The most common ones you are likely to use are shown on this page: <http://www.sunilpatel.co.uk/latex-type/latex-math-symbols/>

You can use this page as a reference or crib sheet, the symbols are rendered as large, high quality images so you can quickly find the L<sup>A</sup>T<sub>E</sub>X command for the symbol you need.

### 2.2.4 L<sup>A</sup>T<sub>E</sub>X on a Mac

The L<sup>A</sup>T<sub>E</sub>X distribution is available for many systems including Windows, Linux and Mac OS X. The package for OS X is called MacTeX and it contains all the applications you need – bundled together and pre-customized – for a fully working L<sup>A</sup>T<sub>E</sub>X environment and work flow.

MacTeX includes a custom dedicated L<sup>A</sup>T<sub>E</sub>X editor called TeXShop for writing your '**.tex**' files and BibDesk: a program to manage your references and create your bibliography section just as easily as managing songs and creating playlists in iTunes.

## 2.3 Getting Started with this Template

If you are familiar with L<sup>A</sup>T<sub>E</sub>X, then you should explore the directory structure of the template and then proceed to place your own information into the *THESIS INFORMATION* block of the **main.tex** file. You can then modify the rest of this file to your unique specifications based on your degree/university. Section ?? on page ?? will help you do this. Make sure you also read section ?? about thesis conventions to get the most out of this template.

If you are new to L<sup>A</sup>T<sub>E</sub>X it is recommended that you carry on reading through the rest of the information in this document.

Before you begin using this template you should ensure that its style complies with the thesis style guidelines imposed by your institution. In most cases this template style and layout will be suitable. If it is not, it may only require a small change to bring the template in line with your institution's recommendations. These modifications will need to be done on the **MastersDoctoralThesis.cls** file.

## Chapter 3

# Mobile Application Development

### 3.1 Mobile Platforms

According to a market share study by IDC (Chau, Govindaraj, and Reith, 2017), the smartphone market is currently clearly dominated by Android, which held over 86.8 percent of the market during the second quarter of 2016. Meanwhile, iOS saw its market share for 2016Q3 grow by 12.7 percent QoQ with 45.5 million shipments. The iPhone 6S followed by its newest model, the iPhone 7 were the best-selling models this quarter. Windows Phone experienced a QoQ decline of 35.2 percent with a total of 974.4 thousand units shipped this quarter. With Microsoft's focus on business users, the decline in the consumer market is expected to continue.

The market shares of the top three mobile platforms and the remaining market during the second quarter of each year between 2015 and 2016 are shown in (e.g. Table 3.1).

#### 3.1.1 Android

Initially developed by Android Inc., which Google bought in 2005, Android was unveiled in 2007, along with the founding of the Open Handset Alliance – a consortium of hardware, software, and telecommunication companies devoted to advancing open standards for mobile devices. Beginning with the first commercial Android device in September 2008, the operating system has gone through multiple major releases, with the current version being 7.0 "Nougat", released in August 2016. Android applications ("apps") can be downloaded from the Google Play store, which features over 2.7 million apps as of February 2017. Android's source code is released by Google under an open source license, although most Android devices ultimately ship with a combination of free and open source and proprietary software, including proprietary software required for accessing Google services. Android is popular with technology companies that require a ready-made, low-cost and customizable operating system for high-tech devices.

TABLE 3.1: Smartphone Operating System market shares from years 2015 to 2016.

Period	Android	iOS	Windows Phone	Others
2015Q4	79.6%	18.7%	1.2%	0.5%
2016Q1	83.5%	15.4%	0.8%	0.4%
2016Q2	87.6%	11.7%	0.4%	0.3%
2016Q3	86.8%	12.5%	0.3%	0.4%

Android is built on a modified

Linux 2.6 series kernel that provides core system services such as security, memory management, process management, network stack and driver model. The kernel and low level tools are contained in the bottom layer, colored red in the illustration. The basic libraries included in Android are programmed in C and C++, and are accessed through the Android application framework.

The Android runtime contains a set of Java core libraries and the Dalvik virtual machine (VM). The Dalvik VM executes

classes in Dalvik Executable (.dex) format, usually transformed from Java byte code to Dalvik byte code.[16 TBD]

Every Android application runs in its own process with its own sandboxed instance of Dalvik VM. abstraction of the underlying hardware for the rest of the software stack.[23 TBD]

The application framework layer gives the developers access to the same framework Application Programming Interfaces (API) used by the core applications. [43] The frameworks are written in Java and provide abstractions the Android libraries and the features of the Dalvik VM.[16]

Applications for Android are developed through the Android Software Development Kit (SDK), usually with the Java programming language. The SDK provides the API libraries and developer tools for building, testing and debugging for Android. Development can be done in any of the current major operating systems and an integrated development environment (IDE) of choice, although Google recommends using Android Studio. Another common option is to use Eclipse IDE with Android Developer Tools (ADT) plugin, provided by Google, which integrates the Android SDK into Eclipse. The ADT allows the developer to test the application with an Android emulator or a connected device, and provides a graphical editor for building the user interface (UI) of the application.[5]

The main distribution channel for Android applications is the Google Play Store, formerly known as Android Market, where developers can publish their applications after registration.

### 3.1.2 iOS

iOS is a mobile operating system created and developed by Apple Inc. exclusively for its hardware. It is the operating system that presently powers many of the company's mobile devices, including the iPhone, iPad, and iPod Touch. It is the second most popular mobile operating system globally after Android. iPad tablets are also the second most popular, by sales, against Android since 2013.[9]

Originally unveiled in 2007 for the iPhone, iOS has been extended to support other Apple devices such as the iPod Touch (September 2007) and the iPad (January 2010). As of January 2017, Apple's App Store contains more than 2.2 million iOS applications, 1 million of which are native for iPads. These mobile apps have collectively been downloaded more than 130 billion times.

The iOS, there are four abstraction layers: the Core OS, Core Services, Media, and Cocoa Touch layers.

The various core frameworks are written in the Objective-C programming language. The Core OS layer contains the low-level features most of the other technologies are built upon. Applications rarely use these technologies directly, but rather use them through the other frameworks. However, the layer contains frameworks for features such as security, Bluetooth support and communicating with external hardware, that can be used by applications if needed. The layer also contains the kernel environment, drivers and low-level UNIX interfaces of the operating system. Access to the kernel and drivers is restricted to a limited set of system frameworks and applications.[9]

The various system services used by applications are contained in the Core Services layer. This includes technologies to support features like location, iCloud storage, peer-to-peer services and networking. The Media layer above it contains the graphics, audio and video technologies needed to implement multimedia features in applications. Finally in the top layer resides the Cocoa Touch framework, providing the key frameworks for building iOS applications. This includes high-level programming interfaces for making animations, networking and modifying the appearance of the application. Cocoa Touch also handles touch-based inputs and multitasking.[8]

Building iOS applications requires using Apple's Xcode IDE on a Mac computer running OS X 10.8 or later and iOS SDK. Xcode provides the standard tools to code, debug and design the interface for the applications. Generally iOS applications are written in Objective-C language.[8]

Applications for iOS are distributed to consumers exclusively through Apple's App Store. Developers enroll in Apple Developer Program and pay yearly fee to be able to publish applications in the App Store, and applications go through an approval process by Apple before appearing in the store.[8] The approval process causes longer development times, but lowers

the number of low-quality applications in the store.[17] The approval process can pose a challenge for applications developed with various cross-platform methods. For example, in 2010 Apple maintained that apps must be "originally written in Objective-C, C, C++ or JavaScript" to be accepted into the store. The restrictions have been eased since then, but applications still sometimes get rejected for being too slow or not feeling native enough. Apple App Store can also reject apps that download executable code or interpret code not contained within the application archive.[32]

### 3.1.3 Windows

Windows 10 is a personal computer operating system developed and released by Microsoft as part of the Windows NT family of operating systems. It was officially unveiled in September 2014 following a brief demo at Build 2014. The first version of the operating system entered a public beta testing process in October, leading up to its consumer release on July 29, 2015.[9]

Windows 10 introduces what Microsoft described as "universal apps" expanding on Metro-style apps. The first release of Windows 10 also introduces a virtual desktop system, a window and desktop management feature called Task View, the Microsoft Edge web browser, support for fingerprint and face recognition login, new security features for enterprise environments, and DirectX 12 and WDDM 2.0 to improve the operating system's graphics capabilities for games.

Universal Windows Platform (UWP) apps[1] (formerly Windows Store apps and Metro-style apps)[2] are apps that can be used across all compatible Microsoft Windows devices, including personal computers (PCs), tablets, smartphones, Xbox One, Microsoft HoloLens, and Internet of Things. UWP apps are primarily purchased and downloaded via the Windows Store.[3]

### 3.1.4 Tizen

Tizen is an open and flexible operating system built from the ground up to address the needs of all stakeholders of the mobile and connected device ecosystem, including device manufacturers, mobile operators, application developers and independent software vendors (ISVs). Tizen is developed by a community of developers, under open source governance, and is open to all members who wish to participate.

The Tizen operating system comes in multiple profiles to serve different industry requirements. The current Tizen profiles are Tizen IVI (in-vehicle infotainment), Tizen Mobile, Tizen TV, and Tizen Wearable. In addition to that, as of Tizen 3.0, all profiles are built on top of a common, shared infrastructure called Tizen Common.

With Tizen, a device manufacturer can begin with one of these profiles and modify it to serve their own needs, or use the Tizen Common base to develop a new profile to meet the memory, processing and power requirements of any device and quickly bring it to market.

Mobile operators can work with device partners to customize the operating system and user experience to meet the needs of specific customer segments or demographics.

For application developers and ISVs, Tizen offers the power of native application development with the flexibility of unparalleled HTML5 support. Tizen also offers the potential for application developers to extend their reach to new "smart devices" running Tizen, including wearables, consumer electronics (TVs, gaming consoles, DVRs, etc.), cars and appliances.

The Tizen project resides within the Linux Foundation and is governed by a Technical Steering Group. The Technical Steering Group is the primary decision-making body for the open source project, with a focus on platform development and delivery, along with the formation of working groups to support device verticals.



The Tizen Association has been formed to guide the industry role of Tizen, including gathering of requirements, identification and facilitation of service models, and overall industry marketing and education.

Tizen provides application development tools based on the JavaScript libraries jQuery and jQuery Mobile. Since version 2.0, a C++ native application framework is also available, based on an Open Services Platform from the Bada platform.

Samsung Releases New Preview of Visual Studio Tools for Tizen

## 3.2 Mobile Applications

Mobile applications are consist of software/set of program that runs on a mobile device and perform certain tasks for the user. Mobile application is a new and fast developing Segment of the global Information and Communication Technology.

Mobile Screens are small, But Mobile apps are big, and life as we know it is on its head again. In a world that's increasingly social and open, mobile apps play a vital role, and Today we have changed the focus from what's on the Web, to the apps on our mobile device. Mobile apps are very imperative. But where do we start? There are many factors that play a part in your mobile strategy, such as your team's development skills, required device functionality, the importance of security, offline capability, interoperability, etc., that must be taken into account. Finally it's not just a question of what mobile application will do, but how we will reach there. get it there.

Mobile applications come in two distinct formats: native apps and web apps. Due to differences in their underlying technology, each approach has inherent advantages and drawbacks.

### 3.2.1 Native Applications

A native mobile app is built specifically for a particular device and its operating system. Unlike a web app that is accessed over the internet, a native app is downloaded from a web app store and installed on the device. Native apps are written in Java, Objective C, or some other programming language. This is changing with HTML5, but functionality is inconsistent and incomplete.

Native apps have a major advantage over web applications the ability to leverage device-specific hardware and software. This means that native apps can take advantage of the latest technology available on mobile devices and can integrate with on-board apps such as the calendar, contacts, and email. However, this is a double-edged sword: while mobile technology is wildly popular, it is also constantly changing and highly fragmented. This makes the task of keeping up with the pace of emerging technology onerous and costly, especially on multiple platforms.

TABLE 3.2: Native App Development across different platform

	Android	iOS	Windows 10	Tizen
<b>Programming languages</b>	JAVA	Objective-C,Swift	XAML,C#	C,C++
<b>Integrated Development Environment</b>	Android Studio	Xcode	Visual Studio	Tizen Studio
<b>Software Development Kit</b>	Android SDK	iOS SDK	Windows SDK	Tizen SDK
<b>App Distribution</b>	Google play	Appstore	Windows Store	Tizen Store
<b>Community</b>	Very Good	Very Good	Good	Limited

### 3.2.2 Mobile Web Apps

A mobile web app is a web application formatted for smartphones and tablets, and accessed through the mobile device's web browser. Like a traditional web application, a mobile web app is built with three core technologies: HTML (defines static text and images), CSS (defines style and presentation), and JavaScript (defines interactions and animations).

Since web apps are browser-based, they're intended to be platform and device independent, able to run on any web-enabled smartphone or tablet. A mobile web app is normally downloaded from a central web server each time it is run, although apps built using HTML5 (described below) can also run on the mobile device for offline use.

However, significant limitations, especially for enterprise mobile, are offline storage and security. While you can implement a semblance of offline capability by caching files on the device, it just isn't a very good solution. Although the underlying database might be encrypted, it's not as well segmented as a native keychain encryption that protects each app with a developer certificate. Also, if a web app with authentication is launched from the desktop, it will require users to enter their credentials every time the app it is sent to the background. This is a lousy experience for the user. In general, implementing even trivial security measures on a native platform can be complex tasks for a mobile Web developer. Therefore, if security is of the utmost importance, it can be the deciding factor on which mobile technology you choose.

### 3.2.3 Hybrid Mobile Applications

Hybrid development combines the best (or worst) of both the native and HTML5 worlds. We define hybrid as a web app, primarily built using HTML5 and JavaScript, that is then wrapped inside a thin native container that provides access to native platform features. PhoneGap is an example of the most popular container for creating hybrid mobile apps.

For the most part, hybrid apps provide the best of both worlds. Existing web developers that have become gurus at optimizing JavaScript, pushing CSS to create

beautiful layouts, and writing compliant HTML code that works on any platform can now create sophisticated mobile applications that don't sacrifice the cool native capabilities. In certain circumstances, native developers can write plugins for tasks like image processing, but in cases like this, the devil is in the details.

On iOS, the embedded web browser or the UIWebView is not identical to the Safari browser. While the differences are minor, they can cause debugging headaches. That's why it pays off to invest in popular frameworks that have addressed all of the limitations.

You know that native apps are installed on the device, while HTML5 apps reside on a Web server, so you might be wondering if hybrid apps store their files on the device or on a server? Yes. In fact there are two ways to implement a hybrid app.

- **Local** - You can package HTML and JavaScript code inside the mobile application binary, in a manner similar to the structure of a native application. In this scenario you use REST APIs to move data back and forth between the device and the cloud.
- **Server** - Alternatively you can implement the full web application from the server (with optional caching for better performance), simply using the container as a thin shell over the UIWebView.

### 3.2.4 Conclusion

Mobile development is a constantly moving target. Every six months, there's a new mobile operating system, with unique features only accessible with native APIs. The containers bring those to hybrid apps soon thereafter, with the web making tremendous leaps every few years. Based on current technology, one of the scenarios examined in this article is bound to suit your needs. Let's sum those up in the following table:

## 3.3 Cross-Platform Development Tools

### 3.3.1 PhoneGap

PhoneGap is the open source framework that gets you building amazing mobile apps using web technology.

Developing your mobile app using HTML, CSS and Javascript doesn't mean that you have to give up on native functionality that makes mobile devices so extraordinary. PhoneGap gives you access to all of the native device APIs (like camera, GPS, accelerometer and more), so that the app you build with web tech behaves just like a native app. but still have some limitations.

TABLE 3.3: The Mobile App Comparison Chart: Native vs. Mobile Web vs. Hybrid

	Native	HTML5	Hybrid
<b>App Features</b>			
Graphics	Native APIs	HTML, Canvas, SVG	HTML, Canvas, SVG
Performance	Fast	Slow	Slow
Native look and feel	Native	Emulated	Emulated
Distribution	Appstore	Web	Appstore
<b>Device Access</b>			
Camera	Yes	No	Yes
Notifications	Yes	No	Yes
Contacts, calendar	Yes	No	Yes
Offline storage	Secure file storage	Shared SQL	Secure file, shared SQL
Geolocation	Yes	Yes	Yes
<b>Gestures</b>			
Swipe	Yes	Yes	Yes
Pinch, spread	Yes	No	Yes
<b>Connectivity</b>			
Connectivity	Online and offline	online	Online and offline
<b>Developer Skills</b>			
language	Objective C, Java	HTML5, CSS, Javascript	HTML5, CSS, Javascript

Since the front end of the application is built in JavaScript, it causes a number of limitations.

- **Data processing:** Native languages are much faster than JavaScript for data processing on the device.
- **Background processing:** A large number of applications rely on background threads to provide a smooth user experience: calculating the GPS positions in the background, for example. PhoneGap APIs are built using JavaScript which is not multi-threaded and hence do not support background processing.
- **Access advanced native functionality:** A number of native APIs are not yet supported by PhoneGap's APIs.
- **Complex Business Logic:** A number of applications such as enterprise applications are quite complex. In this scenario it is simply better to have a certain amount of native code.
- **Advanced Graphics:** Advanced Graphics: Apps that use advanced graphics which can only be accessed using third-party libraries are best done natively.

### 3.3.2 Titanium

The Titanium SDK helps you to build native cross-platform mobile application using JavaScript and the Titanium API, which abstracts the native APIs of the mobile platforms. Titanium empowers you to create immersive, full-featured applications, featuring over 80% code reuse across mobile apps. Appcelerator licenses Titanium under the Apache 2 license and is free for both personal and commercial use.

Titanium's unique trait among the various available cross-platform mobile solutions is that it creates truly native apps. This is in contrast to web-based solutions that deliver functionality via an enhanced web view. Titanium, not wanting to be limited by the native web view, has engaged in a much deeper integration with the underlying platforms. This gives Titanium developers the ability to leverage native UI components and services, as well as near-native performance.

Since the front end of the application is built in JavaScript, it causes a number of limitations.

- **Increasing complexity:** The development complexities (and costs) rise more than proportionally to application complexity increases. The more complex your applications become, the more often you'll have to deal with, on the one hand technical issues (random crashes, weird behaviors, annoying bugs, etc.), on the other hand a greater effort (code organization, MVC separation, multi-device support, multi-platform support, code readability, etc.)
- **No Freemium:** Appcelerator provides StoreKit, a module to enable In-App Purchase to Apple's App Store, but it's a pain. Buggy, poorly documented and it seems to work only partially. Too unstable for production use. Having to renounce the freemium pricing model (apps that are free to download, but require an in-app purchase to be expanded) is not just a minor inconvenience since 72% of total App Store revenue comes from apps featuring in-app purchases.
- **Toolkit pain:** At first there was only Titanium Developer but since last June there is Titanium Studio, an Eclipse-based IDE built on a modified version of Aptana that allows you to manage projects, test your mobile apps in the simulator or on device and automate app packaging. First of all, I sincerely hate Eclipse, yeah, Eclipse is free and the best open source IDE there is, but offers a very poor IDE experience. Most importantly Titanium Studio can go "crazy", encounter weird glitches, stop printing console messages, but the worst thing is when the build process start to ignore changes. You have to continually clean your project every time you make changes or restart with a brand new project. A productivity tool that is uncomfortable and unstable is not a productivity tool and a development tool that is unproductive is not a development tool.
- **Flexibility limitations:** All that glitters is not gold. At beginning you'll love the well-defined Titanium API and you will probably love it even more every time you discover a simple property to enable behaviors that would require several lines of code on Xcode. But sooner or later you will face strange bugs and limitations. For example, if you want to apply a cell background gradient to a grouped table (a very common and easy task with Objective-C) you get that the grouped table becomes plain and the gradient color makes the table slow when scrolling, and you will have to use images... So at first you will save a lot of time but as more complex the project grows you'll lose the saved time in fixing and workarounds.
- **Laggy:** Obviously you can have the most smooth, fast and comfortable user experience possible only with apps developed with a native development environment. This is an obvious observation, but which cannot be omitted. Keep in mind that a Titanium application is the result of an automatic conversion

process from web code to native code. Animations are noticeably laggy and apps are not responsive when return from the background. This is particularly evident with Android devices, less evident with iOS devices (especially those with A5 processor).

### 3.3.3 Xamarin

Launched in 2011, Xamarin is a mono framework used for cross-platform app development. Xamarin brings open source .NET to mobile development, enabling every developer to build truly native apps for any device in C# and F#. We're excited for your contributions in continuing our mission to make it fast, easy, and fun to build great mobile apps.

Xamarin is best for building applications using C# programming language running on .NET Common Language Infrastructure (CLI) (often called Microsoft .NET).

Over 1.5 million developers were using Xamarin's products in more than 120 countries around the world as of May 2015. It is widely used for communicating with the Application Program Interface (API) of common mobile device functions like contacts, camera, and geolocation for android, iOS, and Windows operating systems. It allows developers to use almost 100 % native libraries of both Android and iOS. With a C#-shared codebase, developers can use Xamarin tools to write native Android, iOS, and Windows apps with native user interfaces and share code across multiple platforms, including Windows and macOS.

On February 24, 2016, Microsoft announced it had signed a definitive agreement to acquire Xamarin.

## Chapter 4

# Introduction to Cross-platform Development with Xamarin

Part 1 - Understanding the Xamarin Mobile Platform Part 2 - Architecture Part 3 - Setting Up A Xamarin Cross Platform Solution Part 4 - Dealing with Multiple Platforms Part 5 - Practical Code Sharing Strategies Part 6 - Testing and App Store Approvals

### 4.1 Understanding the Xamarin Mobile Platform

Welcome to this L<sup>A</sup>T<sub>E</sub>X Thesis Template, a beautiful and easy to use template for writing a thesis using the L<sup>A</sup>T<sub>E</sub>X typesetting system.

If you are writing a thesis (or will be in the future) and its subject is technical or mathematical (though it doesn't have to be), then creating it in L<sup>A</sup>T<sub>E</sub>X is highly recommended as a way to make sure you can just get down to the essential writing without having to worry over formatting or wasting time arguing with your word processor.

L<sup>A</sup>T<sub>E</sub>X is easily able to professionally typeset documents that run to hundreds or thousands of pages long. With simple mark-up commands, it automatically sets out the table of contents, margins, page headers and footers and keeps the formatting consistent and beautiful. One of its main strengths is the way it can easily typeset mathematics, even *heavy* mathematics. Even if those equations are the most horribly twisted and most difficult mathematical problems that can only be solved on a super-computer, you can at least count on L<sup>A</sup>T<sub>E</sub>X to make them look stunning.

### 4.2 Architecture

L<sup>A</sup>T<sub>E</sub>X is not a WYSIWYG (What You See is What You Get) program, unlike word processors such as Microsoft Word or Apple's Pages. Instead, a document written for L<sup>A</sup>T<sub>E</sub>X is actually a simple, plain text file that contains *no formatting*. You tell L<sup>A</sup>T<sub>E</sub>X how you want the formatting in the finished document by writing in simple commands amongst the text, for example, if I want to use *italic text for emphasis*, I write the `\emph{text}` command and put the text I want in italics in between the curly braces. This means that L<sup>A</sup>T<sub>E</sub>X is a "mark-up" language, very much like HTML.

### 4.3 Setting Up A Xamarin Cross Platform Solution

L<sup>A</sup>T<sub>E</sub>X is not a WYSIWYG (What You See is What You Get) program, unlike word processors such as Microsoft Word or Apple's Pages. Instead, a document written for

L<sup>A</sup>T<sub>E</sub>X is actually a simple, plain text file that contains *no formatting*. You tell L<sup>A</sup>T<sub>E</sub>X how you want the formatting in the finished document by writing in simple commands amongst the text, for example, if I want to use *italic text for emphasis*, I write the `\emph{text}` command and put the text I want in italics in between the curly braces. This means that L<sup>A</sup>T<sub>E</sub>X is a “mark-up” language, very much like HTML.

### 4.3.1 Code Sharing Options

L<sup>A</sup>T<sub>E</sub>X is not a WYSIWYG (What You See is What You Get) program, unlike word processors such as Microsoft Word or Apple’s Pages. Instead, a document written for L<sup>A</sup>T<sub>E</sub>X is actually a simple, plain text file that contains *no formatting*. You tell L<sup>A</sup>T<sub>E</sub>X how you want the formatting in the finished document by writing in simple commands amongst the text, for example, if I want to use *italic text for emphasis*, I write the `\emph{text}` command and put the text I want in italics in between the curly braces. This means that L<sup>A</sup>T<sub>E</sub>X is a “mark-up” language, very much like HTML.

### 4.3.2 Portable Class Libraries

L<sup>A</sup>T<sub>E</sub>X is not a WYSIWYG (What You See is What You Get) program, unlike word processors such as Microsoft Word or Apple’s Pages. Instead, a document written for L<sup>A</sup>T<sub>E</sub>X is actually a simple, plain text file that contains *no formatting*. You tell L<sup>A</sup>T<sub>E</sub>X how you want the formatting in the finished document by writing in simple commands amongst the text, for example, if I want to use *italic text for emphasis*, I write the `\emph{text}` command and put the text I want in italics in between the curly braces. This means that L<sup>A</sup>T<sub>E</sub>X is a “mark-up” language, very much like HTML.

### 4.3.3 Shared Projects

L<sup>A</sup>T<sub>E</sub>X is not a WYSIWYG (What You See is What You Get) program, unlike word processors such as Microsoft Word or Apple’s Pages. Instead, a document written for L<sup>A</sup>T<sub>E</sub>X is actually a simple, plain text file that contains *no formatting*. You tell L<sup>A</sup>T<sub>E</sub>X how you want the formatting in the finished document by writing in simple commands amongst the text, for example, if I want to use *italic text for emphasis*, I write the `\emph{text}` command and put the text I want in italics in between the curly braces. This means that L<sup>A</sup>T<sub>E</sub>X is a “mark-up” language, very much like HTML.

### 4.3.4 .NET Standard

L<sup>A</sup>T<sub>E</sub>X is not a WYSIWYG (What You See is What You Get) program, unlike word processors such as Microsoft Word or Apple’s Pages. Instead, a document written for L<sup>A</sup>T<sub>E</sub>X is actually a simple, plain text file that contains *no formatting*. You tell L<sup>A</sup>T<sub>E</sub>X how you want the formatting in the finished document by writing in simple commands amongst the text, for example, if I want to use *italic text for emphasis*, I write the `\emph{text}` command and put the text I want in italics in between the curly braces. This means that L<sup>A</sup>T<sub>E</sub>X is a “mark-up” language, very much like HTML.

## 4.4 Dealing with Multiple Platforms

If you are familiar with L<sup>A</sup>T<sub>E</sub>X, then you should explore the directory structure of the template and then proceed to place your own information into the *THESIS INFORMATION* block of the `main.tex` file. You can then modify the rest of this file to



your unique specifications based on your degree/university. Section ?? on page ?? will help you do this. Make sure you also read section ?? about thesis conventions to get the most out of this template.

If you are new to L<sup>A</sup>T<sub>E</sub>X it is recommended that you carry on reading through the rest of the information in this document.

Before you begin using this template you should ensure that its style complies with the thesis style guidelines imposed by your institution. In most cases this template style and layout will be suitable. If it is not, it may only require a small change to bring the template in line with your institution's recommendations. These modifications will need to be done on the **MastersDoctoralThesis.cls** file.

## 4.5 Practical Code Sharing Strategies

If you are familiar with L<sup>A</sup>T<sub>E</sub>X, then you should explore the directory structure of the template and then proceed to place your own information into the *THESIS INFORMATION* block of the **main.tex** file. You can then modify the rest of this file to your unique specifications based on your degree/university. Section ?? on page ?? will help you do this. Make sure you also read section ?? about thesis conventions to get the most out of this template.

If you are new to L<sup>A</sup>T<sub>E</sub>X it is recommended that you carry on reading through the rest of the information in this document.

Before you begin using this template you should ensure that its style complies with the thesis style guidelines imposed by your institution. In most cases this template style and layout will be suitable. If it is not, it may only require a small change to bring the template in line with your institution's recommendations. These modifications will need to be done on the **MastersDoctoralThesis.cls** file.

## 4.6 Dealing with Multiple Platforms

If you are familiar with L<sup>A</sup>T<sub>E</sub>X, then you should explore the directory structure of the template and then proceed to place your own information into the *THESIS INFORMATION* block of the **main.tex** file. You can then modify the rest of this file to your unique specifications based on your degree/university. Section ?? on page ?? will help you do this. Make sure you also read section ?? about thesis conventions to get the most out of this template.

If you are new to L<sup>A</sup>T<sub>E</sub>X it is recommended that you carry on reading through the rest of the information in this document.

Before you begin using this template you should ensure that its style complies with the thesis style guidelines imposed by your institution. In most cases this template style and layout will be suitable. If it is not, it may only require a small change to bring the template in line with your institution's recommendations. These modifications will need to be done on the **MastersDoctoralThesis.cls** file.

## 4.7 Cross-Platform User Interfaces with Xamarin.Forms

If you are familiar with L<sup>A</sup>T<sub>E</sub>X, then you should explore the directory structure of the template and then proceed to place your own information into the *THESIS INFORMATION* block of the **main.tex** file. You can then modify the rest of this file to your unique specifications based on your degree/university. Section ?? on page ??

will help you do this. Make sure you also read section ?? about thesis conventions to get the most out of this template.

### 4.7.1 eXtensible Application Markup Language (XAML)

If you are new to  $\LaTeX$  it is recommended that you carry on reading through the rest of the information in this document.

Before you begin using this template you should ensure that its style complies with the thesis style guidelines imposed by your institution. In most cases this template style and layout will be suitable. If it is not, it may only require a small change to bring the template in line with your institution's recommendations. These modifications will need to be done on the **MastersDoctoralThesis.cls** file.

## 4.8 Testing and App Store Approvals

If you are familiar with  $\LaTeX$ , then you should explore the directory structure of the template and then proceed to place your own information into the *THESIS INFORMATION* block of the **main.tex** file. You can then modify the rest of this file to your unique specifications based on your degree/university. Section ?? on page ?? will help you do this. Make sure you also read section ?? about thesis conventions to get the most out of this template.

If you are new to  $\LaTeX$  it is recommended that you carry on reading through the rest of the information in this document.

Before you begin using this template you should ensure that its style complies with the thesis style guidelines imposed by your institution. In most cases this template style and layout will be suitable. If it is not, it may only require a small change to bring the template in line with your institution's recommendations. These modifications will need to be done on the **MastersDoctoralThesis.cls** file.

Testing and App Store Approvals

### 4.8.1 About this Template

This  $\LaTeX$  Thesis Template is originally based and created around a  $\LaTeX$  style file created by Steve R. Gunn from the University of Southampton (UK), department of Electronics and Computer Science. You can find his original thesis style file at his site, here: <http://www.ecs.soton.ac.uk/~srg/softwaretools/document/templates/>

Steve's **ecsthesis.cls** was then taken by Sunil Patel who modified it by creating a skeleton framework and folder structure to place the thesis files in. The resulting template can be found on Sunil's site here: <http://www.sunilpatel.co.uk/thesis-template>

Sunil's template was made available through <http://www.LaTeXTemplates.com> where it was modified many times based on user requests and questions. Version 2.0 and onwards of this template represents a major modification to Sunil's template and is, in fact, hardly recognisable. The work to make version 2.0 possible was carried out by Vel and Johannes Böttcher.

## Chapter 5

# Introduction to Microsoft azure

### 5.1 Connected Services in Xamarin Studio

Welcome to this L<sup>A</sup>T<sub>E</sub>X Thesis Template, a beautiful and easy to use template for writing a thesis using the L<sup>A</sup>T<sub>E</sub>X typesetting system. If you are writing a thesis (or will be in the future) and its subject is technical or mathematical (though it doesn't have to be), then creating it in L<sup>A</sup>T<sub>E</sub>X is highly recommended as a way to make sure you can just get down to the essential writing without having to worry over formatting or wasting time arguing with your word processor.

L<sup>A</sup>T<sub>E</sub>X is easily able to professionally typeset documents that run to hundreds or thousands of pages long. With simple mark-up commands, it automatically sets out the table of contents, margins, page headers and footers and keeps the formatting consistent and beautiful. One of its main strengths is the way it can easily typeset mathematics, even *heavy* mathematics. Even if those equations are the most horribly twisted and most difficult mathematical problems that can only be solved on a super-computer, you can at least count on L<sup>A</sup>T<sub>E</sub>X to make them look stunning.

### 5.2 Azure App Services

L<sup>A</sup>T<sub>E</sub>X is not a WYSIWYG (What You See is What You Get) program, unlike word processors such as Microsoft Word or Apple's Pages. Instead, a document written for L<sup>A</sup>T<sub>E</sub>X is actually a simple, plain text file that contains *no formatting*. You tell L<sup>A</sup>T<sub>E</sub>X how you want the formatting in the finished document by writing in simple commands amongst the text, for example, if I want to use *italic text for emphasis*, I write the `\emph{text}` command and put the text I want in italics in between the curly braces. This means that L<sup>A</sup>T<sub>E</sub>X is a "mark-up" language, very much like HTML.

#### 5.2.1 A (not so short) Introduction to L<sup>A</sup>T<sub>E</sub>X

If you are new to L<sup>A</sup>T<sub>E</sub>X, there is a very good eBook – freely available online as a PDF file – called, "The Not So Short Introduction to L<sup>A</sup>T<sub>E</sub>X". The book's title is typically shortened to just *lshort*. You can download the latest version (as it is occasionally updated) from here: <http://www.ctan.org/tex-archive/info/lshort/english/lshort.pdf>

It is also available in several other languages. Find yours from the list on this page: <http://www.ctan.org/tex-archive/info/lshort/>

It is recommended to take a little time out to learn how to use L<sup>A</sup>T<sub>E</sub>X by creating several, small 'test' documents, or having a close look at several templates on:

<http://www.LaTeXTemplates.com>

Making the effort now means you're not stuck learning the system when what you *really* need to be doing is writing your thesis.

### 5.2.2 A Short Math Guide for L<sup>A</sup>T<sub>E</sub>X

If you are writing a technical or mathematical thesis, then you may want to read the document by the AMS (American Mathematical Society) called, “A Short Math Guide for L<sup>A</sup>T<sub>E</sub>X”. It can be found online here: <http://www.ams.org/tex/amslatex.html> under the “Additional Documentation” section towards the bottom of the page.

### 5.2.3 Common L<sup>A</sup>T<sub>E</sub>X Math Symbols

There are a multitude of mathematical symbols available for L<sup>A</sup>T<sub>E</sub>X and it would take a great effort to learn the commands for them all. The most common ones you are likely to use are shown on this page: <http://www.sunilpatel.co.uk/latex-type/latex-math-symbols/>

You can use this page as a reference or crib sheet, the symbols are rendered as large, high quality images so you can quickly find the L<sup>A</sup>T<sub>E</sub>X command for the symbol you need.

### 5.2.4 L<sup>A</sup>T<sub>E</sub>X on a Mac

The L<sup>A</sup>T<sub>E</sub>X distribution is available for many systems including Windows, Linux and Mac OS X. The package for OS X is called MacTeX and it contains all the applications you need – bundled together and pre-customized – for a fully working L<sup>A</sup>T<sub>E</sub>X environment and work flow.

MacTeX includes a custom dedicated L<sup>A</sup>T<sub>E</sub>X editor called TeXShop for writing your ‘.tex’ files and BibDesk: a program to manage your references and create your bibliography section just as easily as managing songs and creating playlists in iTunes.

## 5.3 Active Directory Authentication

If you are familiar with L<sup>A</sup>T<sub>E</sub>X, then you should explore the directory structure of the template and then proceed to place your own information into the *THESIS INFORMATION* block of the **main.tex** file. You can then modify the rest of this file to your unique specifications based on your degree/university. Section ?? on page ?? will help you do this. Make sure you also read section ?? about thesis conventions to get the most out of this template.

If you are new to L<sup>A</sup>T<sub>E</sub>X it is recommended that you carry on reading through the rest of the information in this document.

Before you begin using this template you should ensure that its style complies with the thesis style guidelines imposed by your institution. In most cases this template style and layout will be suitable. If it is not, it may only require a small change to bring the template in line with your institution’s recommendations. These modifications will need to be done on the **MastersDoctoralThesis.cls** file.

### 5.3.1 WebAPI

This L<sup>A</sup>T<sub>E</sub>X Thesis Template is originally based and created around a L<sup>A</sup>T<sub>E</sub>X style file created by Steve R. Gunn from the University of Southampton (UK), department of Electronics and Computer Science. You can find his original thesis style

file at his site, here: <http://www.ecs.soton.ac.uk/~srg/softwaretools/document/templates/>

Steve's **ecsthesis.cls** was then taken by Sunil Patel who modified it by creating a skeleton framework and folder structure to place the thesis files in. The resulting template can be found on Sunil's site here: <http://www.sunilpatel.co.uk/thesis-template>

Sunil's template was made available through <http://www.LaTeXTemplates.com> where it was modified many times based on user requests and questions. Version 2.0 and onwards of this template represents a major modification to Sunil's template and is, in fact, hardly recognisable. The work to make version 2.0 possible was carried out by **Vel** and Johannes Böttcher.



## Chapter 6

# Chapter Title Here

### 6.1 Welcome and Thank You

Welcome to this L<sup>A</sup>T<sub>E</sub>X Thesis Template, a beautiful and easy to use template for writing a thesis using the L<sup>A</sup>T<sub>E</sub>X typesetting system.

If you are writing a thesis (or will be in the future) and its subject is technical or mathematical (though it doesn't have to be), then creating it in L<sup>A</sup>T<sub>E</sub>X is highly recommended as a way to make sure you can just get down to the essential writing without having to worry over formatting or wasting time arguing with your word processor.

L<sup>A</sup>T<sub>E</sub>X is easily able to professionally typeset documents that run to hundreds or thousands of pages long. With simple mark-up commands, it automatically sets out the table of contents, margins, page headers and footers and keeps the formatting consistent and beautiful. One of its main strengths is the way it can easily typeset mathematics, even *heavy* mathematics. Even if those equations are the most horribly twisted and most difficult mathematical problems that can only be solved on a super-computer, you can at least count on L<sup>A</sup>T<sub>E</sub>X to make them look stunning.

### 6.2 Learning L<sup>A</sup>T<sub>E</sub>X

L<sup>A</sup>T<sub>E</sub>X is not a WYSIWYG (What You See is What You Get) program, unlike word processors such as Microsoft Word or Apple's Pages. Instead, a document written for L<sup>A</sup>T<sub>E</sub>X is actually a simple, plain text file that contains *no formatting*. You tell L<sup>A</sup>T<sub>E</sub>X how you want the formatting in the finished document by writing in simple commands amongst the text, for example, if I want to use *italic text for emphasis*, I write the `\emph{text}` command and put the text I want in italics in between the curly braces. This means that L<sup>A</sup>T<sub>E</sub>X is a "mark-up" language, very much like HTML.

#### 6.2.1 A (not so short) Introduction to L<sup>A</sup>T<sub>E</sub>X

If you are new to L<sup>A</sup>T<sub>E</sub>X, there is a very good eBook – freely available online as a PDF file – called, "The Not So Short Introduction to L<sup>A</sup>T<sub>E</sub>X". The book's title is typically shortened to just *lshort*. You can download the latest version (as it is occasionally updated) from here: <http://www.ctan.org/tex-archive/info/lshort/english/lshort.pdf>

It is also available in several other languages. Find yours from the list on this page: <http://www.ctan.org/tex-archive/info/lshort/>

It is recommended to take a little time out to learn how to use L<sup>A</sup>T<sub>E</sub>X by creating several, small 'test' documents, or having a close look at several templates on:

<http://www.LaTeXTemplates.com>

Making the effort now means you're not stuck learning the system when what you *really* need to be doing is writing your thesis.



## Chapter 7

# Chapter Title Here

### 7.1 Welcome and Thank You

Welcome to this L<sup>A</sup>T<sub>E</sub>X Thesis Template, a beautiful and easy to use template for writing a thesis using the L<sup>A</sup>T<sub>E</sub>X typesetting system.

If you are writing a thesis (or will be in the future) and its subject is technical or mathematical (though it doesn't have to be), then creating it in L<sup>A</sup>T<sub>E</sub>X is highly recommended as a way to make sure you can just get down to the essential writing without having to worry over formatting or wasting time arguing with your word processor.

L<sup>A</sup>T<sub>E</sub>X is easily able to professionally typeset documents that run to hundreds or thousands of pages long. With simple mark-up commands, it automatically sets out the table of contents, margins, page headers and footers and keeps the formatting consistent and beautiful. One of its main strengths is the way it can easily typeset mathematics, even *heavy* mathematics. Even if those equations are the most horribly twisted and most difficult mathematical problems that can only be solved on a super-computer, you can at least count on L<sup>A</sup>T<sub>E</sub>X to make them look stunning.

### 7.2 Learning L<sup>A</sup>T<sub>E</sub>X

L<sup>A</sup>T<sub>E</sub>X is not a WYSIWYG (What You See is What You Get) program, unlike word processors such as Microsoft Word or Apple's Pages. Instead, a document written for L<sup>A</sup>T<sub>E</sub>X is actually a simple, plain text file that contains *no formatting*. You tell L<sup>A</sup>T<sub>E</sub>X how you want the formatting in the finished document by writing in simple commands amongst the text, for example, if I want to use *italic text for emphasis*, I write the `\emph{text}` command and put the text I want in italics in between the curly braces. This means that L<sup>A</sup>T<sub>E</sub>X is a "mark-up" language, very much like HTML.

#### 7.2.1 A (not so short) Introduction to L<sup>A</sup>T<sub>E</sub>X

If you are new to L<sup>A</sup>T<sub>E</sub>X, there is a very good eBook – freely available online as a PDF file – called, "The Not So Short Introduction to L<sup>A</sup>T<sub>E</sub>X". The book's title is typically shortened to just *lshort*. You can download the latest version (as it is occasionally updated) from here: <http://www.ctan.org/tex-archive/info/lshort/english/lshort.pdf>

It is also available in several other languages. Find yours from the list on this page: <http://www.ctan.org/tex-archive/info/lshort/>

It is recommended to take a little time out to learn how to use L<sup>A</sup>T<sub>E</sub>X by creating several, small 'test' documents, or having a close look at several templates on:

<http://www.LaTeXTemplates.com>

Making the effort now means you're not stuck learning the system when what you *really* need to be doing is writing your thesis.

## Chapter 8

# Chapter Title Here

### 8.1 Welcome and Thank You

Welcome to this L<sup>A</sup>T<sub>E</sub>X Thesis Template, a beautiful and easy to use template for writing a thesis using the L<sup>A</sup>T<sub>E</sub>X typesetting system.

If you are writing a thesis (or will be in the future) and its subject is technical or mathematical (though it doesn't have to be), then creating it in L<sup>A</sup>T<sub>E</sub>X is highly recommended as a way to make sure you can just get down to the essential writing without having to worry over formatting or wasting time arguing with your word processor.

L<sup>A</sup>T<sub>E</sub>X is easily able to professionally typeset documents that run to hundreds or thousands of pages long. With simple mark-up commands, it automatically sets out the table of contents, margins, page headers and footers and keeps the formatting consistent and beautiful. One of its main strengths is the way it can easily typeset mathematics, even *heavy* mathematics. Even if those equations are the most horribly twisted and most difficult mathematical problems that can only be solved on a super-computer, you can at least count on L<sup>A</sup>T<sub>E</sub>X to make them look stunning.

### 8.2 Learning L<sup>A</sup>T<sub>E</sub>X

L<sup>A</sup>T<sub>E</sub>X is not a WYSIWYG (What You See is What You Get) program, unlike word processors such as Microsoft Word or Apple's Pages. Instead, a document written for L<sup>A</sup>T<sub>E</sub>X is actually a simple, plain text file that contains *no formatting*. You tell L<sup>A</sup>T<sub>E</sub>X how you want the formatting in the finished document by writing in simple commands amongst the text, for example, if I want to use *italic text for emphasis*, I write the `\emph{text}` command and put the text I want in italics in between the curly braces. This means that L<sup>A</sup>T<sub>E</sub>X is a "mark-up" language, very much like HTML.

#### 8.2.1 A (not so short) Introduction to L<sup>A</sup>T<sub>E</sub>X

If you are new to L<sup>A</sup>T<sub>E</sub>X, there is a very good eBook – freely available online as a PDF file – called, "The Not So Short Introduction to L<sup>A</sup>T<sub>E</sub>X". The book's title is typically shortened to just *lshort*. You can download the latest version (as it is occasionally updated) from here: <http://www.ctan.org/tex-archive/info/lshort/english/lshort.pdf>

It is also available in several other languages. Find yours from the list on this page: <http://www.ctan.org/tex-archive/info/lshort/>

It is recommended to take a little time out to learn how to use L<sup>A</sup>T<sub>E</sub>X by creating several, small 'test' documents, or having a close look at several templates on:

<http://www.LaTeXTemplates.com>

Making the effort now means you're not stuck learning the system when what you *really* need to be doing is writing your thesis.

### 8.2.2 A Short Math Guide for L<sup>A</sup>T<sub>E</sub>X

If you are writing a technical or mathematical thesis, then you may want to read the document by the AMS (American Mathematical Society) called, "A Short Math Guide for L<sup>A</sup>T<sub>E</sub>X". It can be found online here: <http://www.ams.org/tex/amslatex.html> under the "Additional Documentation" section towards the bottom of the page.

### 8.2.3 Common L<sup>A</sup>T<sub>E</sub>X Math Symbols

There are a multitude of mathematical symbols available for L<sup>A</sup>T<sub>E</sub>X and it would take a great effort to learn the commands for them all. The most common ones you are likely to use are shown on this page: <http://www.sunilpatel.co.uk/latex-type/latex-math-symbols/>

You can use this page as a reference or crib sheet, the symbols are rendered as large, high quality images so you can quickly find the L<sup>A</sup>T<sub>E</sub>X command for the symbol you need.

### 8.2.4 L<sup>A</sup>T<sub>E</sub>X on a Mac

The L<sup>A</sup>T<sub>E</sub>X distribution is available for many systems including Windows, Linux and Mac OS X. The package for OS X is called MacTeX and it contains all the applications you need – bundled together and pre-customized – for a fully working L<sup>A</sup>T<sub>E</sub>X environment and work flow.

MacTeX includes a custom dedicated L<sup>A</sup>T<sub>E</sub>X editor called TeXShop for writing your '**.tex**' files and BibDesk: a program to manage your references and create your bibliography section just as easily as managing songs and creating playlists in iTunes.

## Chapter 9

# Chapter Title Here

### 9.1 Welcome and Thank You

Welcome to this L<sup>A</sup>T<sub>E</sub>X Thesis Template, a beautiful and easy to use template for writing a thesis using the L<sup>A</sup>T<sub>E</sub>X typesetting system.

If you are writing a thesis (or will be in the future) and its subject is technical or mathematical (though it doesn't have to be), then creating it in L<sup>A</sup>T<sub>E</sub>X is highly recommended as a way to make sure you can just get down to the essential writing without having to worry over formatting or wasting time arguing with your word processor.

L<sup>A</sup>T<sub>E</sub>X is easily able to professionally typeset documents that run to hundreds or thousands of pages long. With simple mark-up commands, it automatically sets out the table of contents, margins, page headers and footers and keeps the formatting consistent and beautiful. One of its main strengths is the way it can easily typeset mathematics, even *heavy* mathematics. Even if those equations are the most horribly twisted and most difficult mathematical problems that can only be solved on a super-computer, you can at least count on L<sup>A</sup>T<sub>E</sub>X to make them look stunning.

### 9.2 Learning L<sup>A</sup>T<sub>E</sub>X

L<sup>A</sup>T<sub>E</sub>X is not a WYSIWYG (What You See is What You Get) program, unlike word processors such as Microsoft Word or Apple's Pages. Instead, a document written for L<sup>A</sup>T<sub>E</sub>X is actually a simple, plain text file that contains *no formatting*. You tell L<sup>A</sup>T<sub>E</sub>X how you want the formatting in the finished document by writing in simple commands amongst the text, for example, if I want to use *italic text for emphasis*, I write the `\emph{text}` command and put the text I want in italics in between the curly braces. This means that L<sup>A</sup>T<sub>E</sub>X is a "mark-up" language, very much like HTML.

#### 9.2.1 A (not so short) Introduction to L<sup>A</sup>T<sub>E</sub>X

If you are new to L<sup>A</sup>T<sub>E</sub>X, there is a very good eBook – freely available online as a PDF file – called, "The Not So Short Introduction to L<sup>A</sup>T<sub>E</sub>X". The book's title is typically shortened to just *lshort*. You can download the latest version (as it is occasionally updated) from here: <http://www.ctan.org/tex-archive/info/lshort/english/lshort.pdf>

It is also available in several other languages. Find yours from the list on this page: <http://www.ctan.org/tex-archive/info/lshort/>

It is recommended to take a little time out to learn how to use L<sup>A</sup>T<sub>E</sub>X by creating several, small 'test' documents, or having a close look at several templates on:

<http://www.LaTeXTemplates.com>

Making the effort now means you're not stuck learning the system when what you *really* need to be doing is writing your thesis.

## Chapter 10

# Conclusion

### 10.1 Goal Fulfilment

Welcome to this L<sup>A</sup>T<sub>E</sub>X Thesis Template, a beautiful and easy to use template for writing a thesis using the L<sup>A</sup>T<sub>E</sub>X typesetting system.

If you are writing a thesis (or will be in the future) and its subject is technical or mathematical (though it doesn't have to be), then creating it in L<sup>A</sup>T<sub>E</sub>X is highly recommended as a way to make sure you can just get down to the essential writing without having to worry over formatting or wasting time arguing with your word processor.

L<sup>A</sup>T<sub>E</sub>X is easily able to professionally typeset documents that run to hundreds or thousands of pages long. With simple mark-up commands, it automatically sets out the table of contents, margins, page headers and footers and keeps the formatting consistent and beautiful. One of its main strengths is the way it can easily typeset mathematics, even *heavy* mathematics. Even if those equations are the most horribly twisted and most difficult mathematical problems that can only be solved on a super-computer, you can at least count on L<sup>A</sup>T<sub>E</sub>X to make them look stunning.

### 10.2 Future Work

L<sup>A</sup>T<sub>E</sub>X is not a WYSIWYG (What You See is What You Get) program, unlike word processors such as Microsoft Word or Apple's Pages. Instead, a document written for L<sup>A</sup>T<sub>E</sub>X is actually a simple, plain text file that contains *no formatting*. You tell L<sup>A</sup>T<sub>E</sub>X how you want the formatting in the finished document by writing in simple commands amongst the text, for example, if I want to use *italic text for emphasis*, I write the `\emph{text}` command and put the text I want in italics in between the curly braces. This means that L<sup>A</sup>T<sub>E</sub>X is a "mark-up" language, very much like HTML.





## Appendix A

# Frequently Asked Questions

### A.1 How do I change the colors of links?

The color of links can be changed to your liking using:

```
\hypersetup{urlcolor=red}, or  
\hypersetup{citecolor=green}, or  
\hypersetup{allcolor=blue}.
```

If you want to completely hide the links, you can use:

```
\hypersetup{allcolors=.}, or even better:  
\hypersetup{hidelinks}.
```

If you want to have obvious links in the PDF but not the printed text, use:

```
\hypersetup{colorlinks=false}.
```



# Bibliography

Chau, Melissa, Navina Govindaraj, and Ryan Reith (2017). "Smartphone OS Market Share, 2016 Q3." In: *International Data Corporation Smartphone OS Market Share, 2016 Q3* 72.1, pp. 1–1. URL: <http://www.idc.com/promo/smartphone-market-share/os>.