



edited by  
**Hussain Fakhruddin**

# **The Secret Sauce Behind Successful Apps: A Handbook For Developers**



Visit us on LinkedIn at:  
<https://www.linkedin.com/company/teksmobile>

# About Teksmobile

Teksmobile is one of the fastest growing mobile app development companies in the world. We specialize in the development of custom applications for the Apple (iOS, watchOS, tvOS, macOS) and Android (smartphone, tablet, wearables) platforms. Team Teks was formed in early-2006, and to date, we have successfully created more than 1000 mobile apps.

# **The Secret Sauce Behind Successful Apps: A Guidebook For Beginners**

**Edited by:** Hussain Fakhruddin

---

Copyright © 2017, Teksmobile. All Rights Reserved.

---

This ebook is a proprietary material of Teksmobile. The book as a whole, or any part thereof, cannot be reproduced in any way (scanned, photocopied, typed, etc.), using any electrical or mechanical method. Such act is permitted only after obtaining prior permission of Teksmobile. Infringement of the copyright would amount to a chargeable offence.

---

**Images & Icons:** Readers are also requested to note that all images used in this ebook are the sole property of Teksmobile. None of the visual elements present here can be reproduced, resold, or reused for any commercial purpose. The icons present in this book have been sourced from their corresponding software/tools, with due accreditation.

---

**'The Secret Sauce Behind Successful Apps: A Guidebook For Beginners'** is an exhaustive technical ebook brought to you by Teksmobile, purely for learning and instructional purpose. Please use it as such!

# Contents

Section	Page
Preface	4
<b><u>Chapters</u></b>	
From Ideas To Apps	7
Making A Successful App: The Tricks	24
Storyboarding and App Testing	74
User Experience: 91	
The 15-Seconds-App-Rule	
Mobile Game Development: Tips & Tools	100
Mobile Apps For Kids	115
Optimizing apps At The Store	122
Success Stories In Businesses	130

# Preface

There are apps to keep track of the latest sports scores and news updates. Don't have time to read the newspaper in the morning? No problems...browsing any of the several news apps will do the trick. An otherwise boring bus ride or train journey can be made interesting with mobile games - of which there is a wide gamut of varieties in the app stores. Love to stay connected with your Facebook buddies and the Twitter world? There are plenty of cool, snazzy social networking apps available. With education technology getting increasingly advanced, the number (*and variety*) of learning apps and games for kids is also going up rapidly. Mobile apps have well and truly become 'must-have' companions for our daily lives!

The global mobile app industry is growing...and growing fast. There are more than 6.5 million apps available for download in the different app stores - with Google Play Store leading the way with 2.8 million applications, and Apple App Store taking up the second spot with 2.2 million apps. By 2020, worldwide shipments of smartphones is expected to reach 2.87 billion. According to an App Annie report, the value of the global mobile app industry

will spike to \$188.9 billion by the end of this decade - an increase of almost 270% over the 2015 figure. The stats and figures clearly underline the position of mobile app development as one of the fastest growing segments in the field of technology. The field is intensely competitive - with developers striving to come up with newer, more innovative, and uniformly user-friendly applications to ensure higher download figures. This book aims to familiarize mobile app developers (*both freelance and those working in companies*) with updated, effective strategies to make their apps stand out in the overcrowded app marketplaces.

Good apps come from great app ideas - and we will dwell on that in the initial chapter of this ebook. A set of basic tips and pointers for making your app successful will be presented next. Then, we will turn our collective attentions on the importance of mobile app designing (*graphics, animation, etc.*). A brief section on wireframing and storyboarding later, we will straightaway move on to app testing standards and best practices. The next topic under the spotlight would be mobile apps for kids, and how they should be made. A chapter on game development and popular game engines will follow. We will next devote an entire section on app store optimization strategies. The book rounds off with a few fascinating case studies on how mobile apps have boosted businesses in the real-world.

We trust that you will find the topics covered in this book to be extensive and beneficial.

Happy Reading!

***-- Team Teksmobile***

**Note:** In this ebook, we will be primarily concerned with app-making for the two biggest smartphone platforms - Apple's iOS and Google's Android.

# 1. From Ideas To Apps

---

***In this chapter, you will learn:***

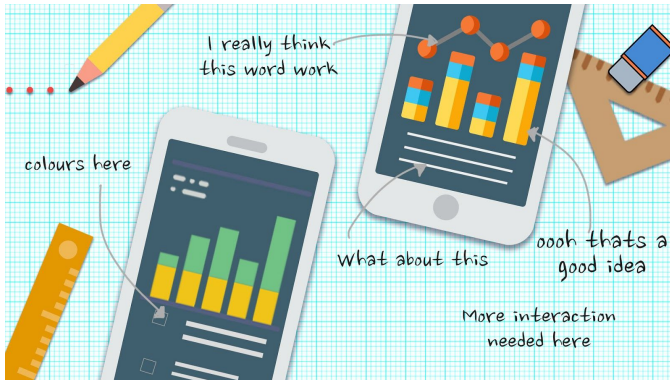
- *The rationale behind mobile app ideas.*
  - *Ways to validate an app idea.*
  - *Some ground rules for transforming ideas into apps.*
- 

Behind every successful mobile application is a solid, viable, and relatively unique idea. The most advanced technical knowledge and coding acumen are of little use, if you do not have a definite idea - about the basic nature and purpose of your app - to start with.

A good, viable mobile app idea can be thought up by practically anyone. The trick lies in being able to correctly judge the merits of each idea, and deciding whether it would be worth investing time, money, and of course, professional expertise on an idea.



## 1.1 How To Come Up With Good App Ideas?



### 1.1.1 The Importance Of ‘Targeting A Problem’

Conduct customer surveys and researches (*email questionnaires and social media interactions are easy tools for these*) to find out about things that people want from their smartphones/tablets – but there are no apps for the same yet. Don’t waste your time on an app idea that seems captivating but is not of much practical use. In the United States, the average smartphone-owner has close to 30 apps installed, and they have little tolerance for ‘*useless*’ stuff. Identify ‘*problems*’ or ‘*gaps*’, and think how a new app can solve it.

### **1.1.2. Unique Ideas Work...But Only When They Are Not Too Outlandish**

Selling public parking places, creating weird emojis, providing chat options based on GPS-functionality – all of these are unique app ideas, and terrible ones as well. Problems can crop up for such ‘*way too different*’ mobile apps on two counts: first, the chances of these apps getting rejected at the stores will remain high, and second, such overly-ambitious apps might have few takers (*after they have been approved*). A worldwide mobile app revolution is on and people are opening up to various types of applications – but to start off with, keep your ideas relatively conventional (*not stereotypical though*).

### **1.1.3. Go Beyond Your Gut Feelings**

You might ‘*feel*’ that a certain type of iPhone or Android app would work. Sadly, such random hunches and gut feelings have every chance of failing. Instead, check out the top-selling charts at the stores – and find out the details of applications that have already become popular. This would give you a starter’s idea about the nature of mobile apps that have maximum chances of succeeding.



person. Do a simple test – gather some of your friends/family-members, and describe your idea to them. If the listeners do not get a proper idea about your proposed app in 3-4 lines, that's a telltale sign that you have overcomplicated things. Abandon that idea, and try to think up something simpler.

### **1.1.6. Base Ideas On Trending Topics**

Google Trends is your best buddy for this. Find out which topics, ideas and tech concepts are trending at any time – and try to make an app that incorporates these popular trends. Be wary of short-lived fads though. Trends would give you a clear picture of what most people are interested in – and your goal is to make apps that satisfy those interests.

### **1.1.7. Strive To Make Improvements**

You might be wondering how you can possibly find an absolutely '*new*' app idea – particularly when there are well over 2 million applications each, at iTunes and the Google Play Store. Don't worry – no one has said that mobile app development requires full-blown innovation. While studying the most downloaded apps from the stores, jot down points on how you can add/improve its features. You will be, in essence, piggybacking an existing app idea

(yay!), but with additional inputs and thoughts of your own.

### **1.1.8 Get Everyone Involved**

The more people contribute inputs for your app ideas, the better. It's all about effective brainstorming – and if you are a professional mobile app entrepreneur, get as many of your employees as possible to provide ideas and concepts. For instance, if you are planning to make a mobile gaming app for kids, ask everyone to submit game ideas. The best ideas can come from the unlikeliest sources – so make sure you tap every resource at your disposal.

Procrastination hampers every business, and mobile app development is no exception to that. Finding and nurturing app ideas is not something only the biggest creative geniuses can do. Even regular developers can zero in on amazing ideas.

## 1.2 Validating App Ideas



Let's face it - a challenge to think up a concept that is unique, and has chances to be actually transformed into an application is not the easiest. That's precisely why it is important to ensure that when you actually get an idea (*and hey, it can come to you at any time, at any place, when you are doing anything!*), you validate it properly.

### 1.2.1. One Idea At A Time

You have four great app ideas, so why not try validating each of them simultaneously and start working on them? Sounds cool on paper, but doing so would be a folly in reality. It is always advisable to process one app idea at a time. If you try to balance multiple ideas, chances are that you will only skim through them – and not be able/will not

have the time to perform in-depth analysis of all the aspects of each idea. And, as we all know, a half-baked app idea is never worth wasting time, money, or effort on. Give your best to one app idea, and move on to another one only when you are done with the first.

### **1.2.2. Be Prepared For ‘Similar’ Apps**

There are 2.8 million apps in Play Store, another 2.2 million at the App Store – and chances are pretty low that your idea (*which you probably consider to be really ‘unique’*) hasn’t already been thought up, worked upon, and released by other mobile app developers. Do not lose heart by this though. Check the apps that are similar to the one you wish to build, list down their features, and go through the user-reviews they have got. Try to find out how you can improve on the existing applications and whether there is a gap that needs to be plugged.

### **1.2.3. Test The Viability Of App Idea**

Just because you are absolutely in love with what you have thought up does not mean that others would be similarly impressed by it too. Professional mobile app developers generally conduct surveys before starting any app development process – but it would be a smart option to test the idea on your

own. Share your idea with your friends, colleagues, neighbors, other acquaintances (*basically, anyone you know who owns a smartphone*). There are plenty of mobile app forums online too, where you can submit your app-idea. If the feedback isn't exactly positive, take the hint – and proceed to your next idea.

#### **1.2.4. 'Good-To-Have Apps' vs 'Must-Have Apps'**

Always go for the latter. There is no harm in conceptualizing a fancy app idea – which, when developed and released, will be an interesting inclusion in the app stores. However, analysts and app developers agree that such '*good-to-have*' apps often fail to motivate general smartphone users to download them. Instead, your idea should be about creating a '*must-have*' app – an app that would solve a particular, important, regular need of users. People should be convinced that having your app on their personal devices would indeed be handy.

#### **1.2.5. Studying App Store Trends Is Vital**

Check the app store of your country, to find out which are the top ranking apps in terms of downloads as well as revenues. By monitoring the



list of top paid apps on a regular basis, you will also get an idea on how to monetize your mobile application. On the other hand, checking the free apps (*which, on average, have nearly ten times the download volume as paid apps*) will familiarize you with viable methods to include in-app purchase options. If the hot trends in the app store have nothing related to the app idea you have, moving on to another concept would be advisable.

### **1.2.6. What Are The Search Volumes?**

It's all very well to come across an app idea that seems nice enough to you – but will the general public, the people who use smartphone applications regularly, be interested in it? The only way to get an objective answer to this is by estimating the expected search volume that your app would generate. There are handy tools like *Google Keyword Planner*, where you can type the nature of your app and/or other words related to your idea. If you find there is enough interest in your concept, hire a good app development company to work on your project. In contrast, if search volumes appear thin, it would be advisable to give that idea a miss and try thinking up another one.

### **1.2.7. Make Use Of Social Media**

Another easy and effective validation method that many mobile app development professionals recommend. Prepare a website/single landing page dedicated to your app idea, optimize it (*hire a digital marketing agency for the purpose, if required*), and monitor the number of hits the page is getting. Analyse the user-behaviour (*bounce rates, average duration of stay, etc.*) as well. These will give you a fair idea about whether people have indeed got hooked to your idea. Similarly, you can use channels like Facebook, Twitter and LinkedIn groups to create promotional campaigns for your application (*yes, even before developers start to work on it*). If there is sufficient interest, consider starting a paid Adwords or Facebook ad campaign.

### **1.2.8. Communicate With Experts; Find What Others Think About Your Idea**

There is no better judge of your app idea than actual people – who are likely to download and use your application. Share the broad details of your concept with people you know and trust (*don't just talk to anyone, for violation of intellectual property rights might be an issue*). Attend conferences and seminars organized by local mobile app entrepreneurs, network with developers, and ask

whether they feel your idea is indeed worth working on or not. These people have years of experience in the app development industry – and they can easily identify a good app idea when they see (*or hear*) one. General people, on the other hand, can tell you whether your idea seems interesting enough.

Creating a blog is a great way to gauge such user-interest levels for free. Publish posts on a regular basis about different aspects of your app idea, and keep a tab on the readership levels and the comments that readers leave on your blog. Once you are convinced about the '*worthiness*' of your app idea - and have evidence to back up your belief - you can move on to actually working on it.

## 1.3. Transforming Ideas Into Apps



The best app ideas can go to waste, if you are not proactive about implementing them. Procrastination is something that needs to be avoided at all costs - and it is important to start working on your idea, as soon as the latter has been properly validated. In many cases, even when the idea is worked on – the end-product comes out to be rather different from what had originally been conceived. A pre-planned, systematic procedure needs to be followed to transform a promising idea into a well-functioning mobile application.

### 1.3.1. Take Calculated Risks

For every aspiring mobile app entrepreneur, being prepared to take calculated risks is an absolute must. There is a tendency among many people to wonder whether their app-idea would be viable,

whether companies would agree to take up the project, and if the revenues from the mobile app would be adequate. There are a lot of things that can go wrong for first-timers, but don't let such setbacks thwart your ambition. Take risks – for the potential payoffs can be huge. Apple app developers worldwide earned more than \$20 billion last year – that should give you an idea.

### **1.3.2. Partner With A Qualified Developer Company**

Visit the iTunes store or the Google Play Store (*depending on whether it is an iOS app or an Android app you wish to develop*), and browse through the featured apps. You will find the company names on those app-pages. Send along your app-idea, and request for a free quote from at least 4-5 companies. Select the one that seems to be competent enough and is offering the best deal. Ideally, go with a company that has expertise in cross-platform mobile app development.

### **1.3.3. Avoid Wasting Time On 'Duplicate Apps'**

Carefully browse through the apps at the App Store/Play store (*it will take some time, for there are over 5 million applications at the stores*

*combined*). Find out if there already exists one or more app that uses the same ‘*core*’ idea as yours. If yes, there is every chance that your app will run into problems during the approval stage. Its popularity levels would also be hit. The key to success lies in having a unique idea, that can be transformed into an app that would have a wide enough target market.

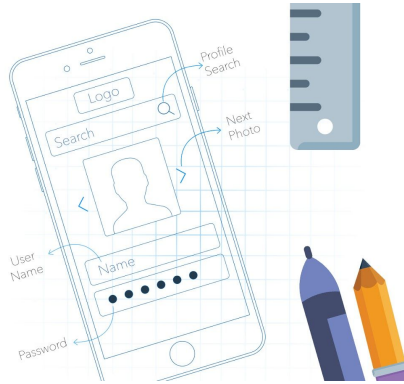
### **1.3.4. Collaborate With The Developers At All Times**

This is where many perfectly good app ideas go awry. Any decent app development company would mention the number of weeks within which your app will be ready – but that does not mean you should contact the firm only AFTER that time-span. Stay in touch with the developers on an ongoing basis, ask for the mockups and prototypes to be shown to you, and if you have any suggestions – do share them. You won’t like it one bit if you had thought up the idea of an ‘*apple*’, and the company hands you an ‘*orange*’. Remember, you are a part of the development team too.

### **1.3.5. Studying A Visual Representation Of The Idea**

Sketch out visual representations of your ideas, to get a feel of how the app (in its final form) would

work. There are several mind-mapping software available as well, which serve the same purpose. Make it a point to reach out for that pen and notepad every time you hit upon a new app idea. The 'look' of your idea matters!



### 1.3.6. Intellectual Property Rights Of Apps

Your idea is, well, your own. Do not let shady app development companies take you for a ride (*and the promise of a free quote and affordable costs is not enough*). Make sure that all the requisite non-disclosure/non-competing agreements are signed by your chosen developer, before the actual work starts. Once the app creation is complete and it is delivered to you, you should be the sole owner of all intellectual property rights on it.

### 1.3.7. Divide The App Development Process In Milestones

Make sure that each milestone is: a) being completed on time and b) being managed according to your preferences. There will be a feel-good feeling as each milestone gets ticked off, and you approach your goal of seeing your idea in the form of a nice and functional app.

### 1.3.8. Test App Prototypes/Betas Carefully

Ideas, by definition, cannot have 'errors'. Software programmers and mobile app developers are humans though, and they can very well make mistakes. Ask about the mobile app testing procedures that the company you have selected follows. A buggy app can ruin a perfectly great app-idea. Get all the bugs fixed before the application is submitted at store(s).

Making a mobile app requires a fine balance - between sitting on an idea for too long (*thereby wasting it*) and being in an almighty rush to complete and launch the product (*compromising on its quality*). A systematic approach is what's required to be able to effectively utilize app ideas, for transforming them into full-fledged applications.

---



## 2. Making A Successful App: The Tricks

---

***In this chapter, you will learn:***

- *The elements of the 'App Triangle'*
  - *App development best practices*
  - *How to deliver optimized user-experience (UX)*
  - *Reasons for frequent app uninstallations*
  - *Strategies to satisfy app clients*
  - *Key elements of a successful app*
- 

The mobile app marketplace is too competitive for taking success as a given. A 2016 report found that nearly 90% of all applications in the Apple App Store fell under the category of 'zombie apps' (*i.e., apps that cannot be near-zero discoverability and cannot be found organically*). These apps - generally with some glitch or the other - are not updated for years, and are practically '*invisible*' at the store. Google Play Store also has a high

percentage of such ‘*zombie apps*’. If you are planning to make a mobile application, you need to ensure that your product stays well clear of this category.

## 2.1. The App Triangle

The app development process involves a three-way trade-off. We can present these three elements - Quality, Cost and Time - as the three corners of a triangle:



The ‘*App Triangle*’ implies that any one of the three elements cannot be improved/more optimized without resulting in a compromise on either or both of the other elements. For instance, if you wish to ensure perfect quality for your app - you will have to invest the required amount of time and money for it. If the time-to-market has to be shortened, the quality aspect - more often than not - suffers. In the

ideal scenario, from the perspective of Economics, an '*App Triangle*' is pareto-optimal.

## 2.2. Best Practices For Making An App



In December 2016, 59355 new apps were submitted at the Apple App Store (*along with 25925 games*). It is easy enough for a new app to get pretty much '*drowned out*' in the crowd - and the onus lies on developers to ensure that a new application indeed has the potential to become successful. For this, some basic guidelines and best practices have to be followed.

## **2.2.1. Pre-development Research & Demand Estimates**

With so many apps – belonging to so many categories – already in existence, it's tricky to stumble upon an entirely unique app idea. Instead of trying to design a concept that would be too novel for its own good, check all the apps that belong to the same/similar genre that you are interested in. Find out what the highest grossing apps are doing well, and how your app can be an improvement over them. In addition, perform a demand analysis for your software separately. Jot down all the keywords closely related to your app, put them on any reliable online keyword research tool (e.g., *Google Keyword Planner*), and find out what their average search volumes are. Unless your app idea belongs to a category that has sufficient demand, developing it won't be worth the resources.

## **2.2.2. The Importance Of Having Core Feature(s)**

Feature mining is one of the most important tasks, during the conceptualization stage of a new iOS/Android application. Resist the temptation of including as many features and functionalities as possible in the app – which would only end up

confusing users. Draw up two lists – of the ‘*must-have*’ and the ‘*nice-to-have*’ features – and include only the former in the introductory version of an application. Your app should have one core purpose, and one feature to fulfill that purpose. Additional features can be included in subsequent updates.

### 2.2.3. Apps Should Be Problem-Solvers



In order to ensure the practical utility of your app, try to think of common problems (and worry not, there are plenty of problems everywhere) that a new mobile application might be able to solve. Form a clear concept about the core function of the app, and the target audience it is going to serve. All innovations are made to solve one problem or another, and a mobile app should not be an

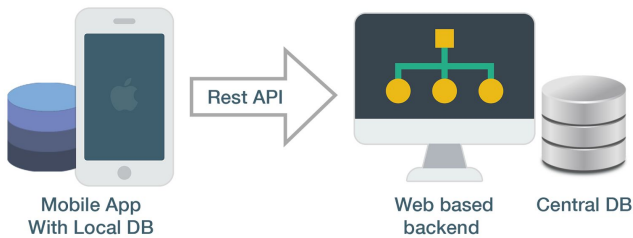
exception. Clarity of thought will help in the creation of the application later.

## **2.2.4. Wireframes & Mockups**

Put yourself in the shoes of the prospective users of your app, and try to decipher their behaviour flow (*user flow research*) and overall in-app navigation. Draw up (*and yes, rough sketches will do*) the screens you would like to have, sequentially...in the manner the users would be likely to move through them. In the wireframe, include all the features that you wish to include in your app. Make sure that there is a logical flow from each screen to the next one. Once the low-fidelity wireframes are ready, create a mockup (*i.e., a high-fidelity visual representation*) of your application. Drawing a flowchart will help you understand the overall function-flow of your application.

The storyboarding technique is extremely useful for getting a proper feel of how the different app screens will be connected to each other.

## 2.2.5. Handling The Backend



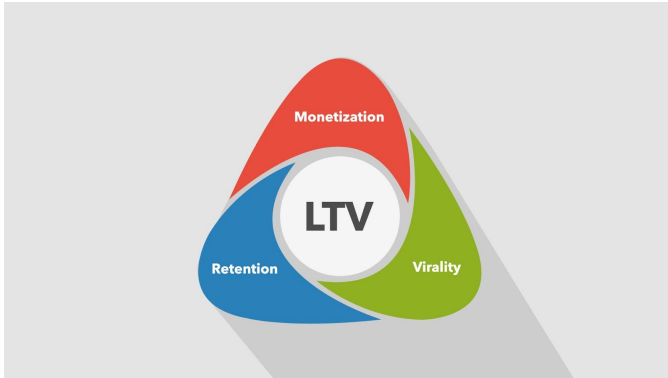
That's where application program interfaces (APIs) come into the picture. You can either create your own APIs, or use the custom APIs already available – depending on the requirements of your application. Efficient APIs facilitate smooth cloud-connectivity of an application – and allow apps to fetch important information, as and when required (*e.g., Uber fetching the location of users with Google Maps API*). In most cases, instead of spending time on creating a custom backend, app makers prefer deploying a mobile backend-as-a-service (*mBaaS*) architecture. Proper utilization of APIs add a lot of efficiency to a mobile app.

## 2.2.6. A Good App Is A Well-Designed App

Brilliant app ideas and high-level coding can all go to waste, if the application is visually sub-par. Remember that software designing is not only about how the app will ‘look’, but it also includes how the technology ‘functions’...what type of end user-experience (UX) it manages to deliver. Use the wireframes you had drawn up earlier, and get high-resolution versions of the same designed (*preferably by an expert graphic designer*). Create the layouts of each screen (*also known as ‘skins’*) very carefully, and always give prime consideration to the convenience of final users. Every tappable element should be properly designed, the main tabs/sections should be easily visible, and the overall in-app navigation must be user-friendly. A mobile app that ‘*seems*’ complicated is almost certain to be discarded.



## 2.2.7. Tracking & Monetization



How many people have downloaded your app? What is the app's average engagement level and what is the behaviour-flow of users while using the application? These are issues that you need to monitor on a continuous basis – and for that, implementing a reliable analytics system is of paramount importance. These metrics can also indicate the scopes of improvement in the application (*that you can fix in later updates*). What's more, your app has to be monetized – either with advertisements or with in-app purchases (IAP). In case you are planning to launch a paid app, do all the essential research to find out what the '*correct app price*' would be.

## 2.2.8. Repeated Rounds Of App Testing

App testing is an iterative process – it has to be done at several successive stages. Such an approach helps in identifying bugs and code mistakes quickly – problems which would have been tough to detect during the time of final testing. In addition to the simulators/emulators available in the iOS and Android integrated development environments (IDEs) respectively, apps also have to be tested on actual devices, by real people. Pay attention to all the feedback and suggestions you receive, and make the necessary changes accordingly. Only after thorough testing and removal of all bugs (*if any*) can your app be considered ready for submission at the stores.

## 2.3. User-End Experience (UX)



On average, 1 out of every 4 downloaded apps do not last even one full day on users' devices. This stat clearly indicates that most applications come up short on the user-experience count. There are plenty of alternatives/competitors available in the app stores - and if a smartphone-owner is not 'satisfied' with the performance of an application, (s)he will simply skip to a rival app. While creating a new app, you need to put yourself in the user's shoes - and make sure that your product delivers optimal UX.

### 2.3.1. User-Interaction vs User-Experience

Although often used synonymously, the two terms have entirely separate meanings. '*User-Interaction*'

refers to the way in which a person uses your application – how (s)he navigates through the screens, exchanges/stores data, performs transactions, and the like. In a nutshell, it is the overall behaviour of the user while operating an app. On the other hand ‘*user-experience*’ is all about whether a person likes or totally hates using the app – maybe due to poorly-conceived layouts and interfaces, or a laggy nature, or frequent crashes, or any of the several other problems if the app developer is not careful enough. User-interaction generates a positive or negative influence in the minds of the user, and that translates to user-experience.

### **2.3.2. Gauge The Target Audience’s Preferences And Design Accordingly**

The role of UI/UX designers, graphic artists and animators in influencing the app-using experience cannot be overemphasized. Prime importance has to be placed on first identifying the profile of the audience group a new app would target, and then instructing the creative department to prepare the layouts and designs accordingly. For instance, a personal finance application should have a completely different look and feel from a mobile fitness tracker, which, in turn, should not be anything like an Android or iPhone app for kids.

Know your users well, and design keeping their requirements and preferences in mind.

### **2.3.3. Use The Screen Real Estates Of Devices With Intelligence**

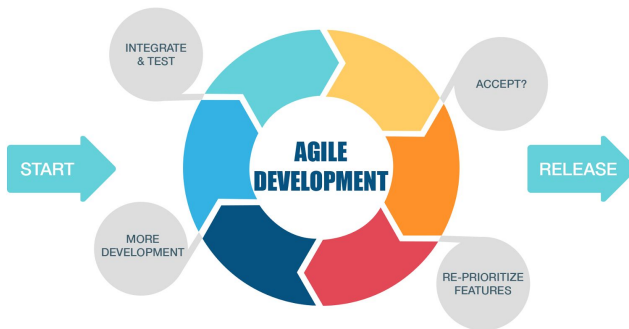
Once you have identified the platform and the devices that your app will be operable on – prepare the screens in a manner that the total space on the device screens is properly utilized. There should neither be any unnecessary clutter (*stay away from cramming in too much of content or images*), nor should there remain too much of blank space. In this context, the job of Android developers is more challenging than their iOS counterparts – simply due to the huge array of Android devices currently available in the market. For Apple, it's only about the iPhone, the iPad, the iPod Touch, and yes, the Watch.

### **2.3.4. Speed Matters**

Speed is a vital factor in determining the success or doom of an application. The average app-interaction time of users is small (*way less than that with desktop apps*), and people want to get things done quickly. After all, that is one of the key reasons for downloading any app. If your app takes too long to load, the splash screen remains visible for more than 10-15 seconds, users have to

navigate through a truckload of screens – rest assured that the application will be chunked out of most devices pretty soon. Follow the simple three-tap rule – a user should not have to make more than three taps after opening an app, to arrive at the screen (s)he wants to view.

### 2.3.5. Agile App Development Cycles...But Nothing Too Rushed



Agile development is something every mobile app company has to follow, while working on new projects. The competition is getting fiercer, and there is a definite need to shorten the app development cycles. However, this should never be done by glossing over the need for implementing top-notch, user-friendly, customized UI/UX designs in an application. Depending on the type of app you are working on, frame an idea about how long it

would take to complete, and specify that clearly when you provide free app quotes to clients. If you try to rush through the development, you will almost surely end up making a defective application – one that would satisfy neither the client nor the end-users. Invest as much time on an app as it warrants.

### **2.3.6. The Battery Consumption Factor**

In 2015, over 95% of downloaded apps were uninstalled within one month. A common complaint among app-users is that certain applications cause significant battery drain and/or lead to devices getting overheated. Make sure that your app is not a battery or bandwidth hog, and it does not adversely affect the overall performance of the target devices in any other way. Many people play games or chat on social networking apps for hours on end – and even such *‘power-users’* should not face any problems.

### **2.3.7. Seamless Software & Hardware Integration**

The mobile app you create should be able to use to the hardware resources of the device(s) it is installed on. Otherwise, the functionality of the application will always remain half-baked, and it

won't take long for a user to find a better alternative. In addition, developers also need to make their apps well-integrated with other native applications on devices. That's the only way smartphone-owners can be given the holistic mobile-using experience that they often seek. For instance, an image-based app should be able to integrate the camera features of a phone.

### **2.3.8. Collect User Feedback. Work On Them**

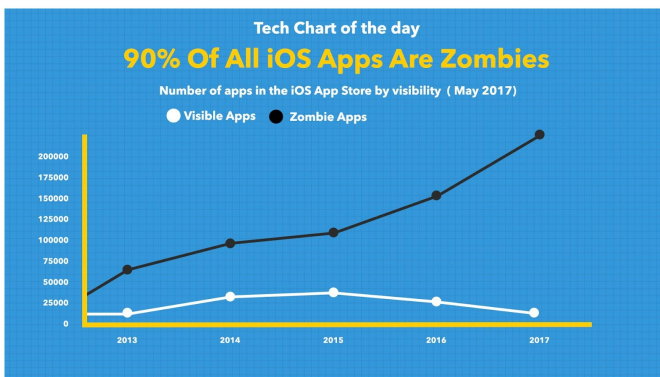
If a mobile app developer hopes to create successful apps, (s)he needs to be open to feedback and suggestions. Share the wireframe sketches and mockups with the potential users, and find out whether they have any recommendations or improvement suggestions that can be implemented. For mobile app development agencies, seeking feedback from the clients at every stage is imperative. Even the best mobile app developers can overlook certain stuff – and the opinions and feedback from third-party sources always come in handy.

The presence of repetitive, invasive ads - which might cover a section of the actual app screens - are a major turn-off. If your app monetization strategy does include the placement of in-app ads, make sure that the advertisements are not



disruptive or, in any way, inappropriate. It is also imperative to have a robust customer service team on hand - to address the problems users might face.

## 2.4. A Closer Look Into The ‘World Of Zombie Apps’



The enormous (and constantly increasing) volume of ‘zombie apps’ points to one thing - users do not like the app they have downloaded, and hence do not hesitate to delete it soon enough. This is a shame - since the app must have done well on the first point of interaction (PoI), when the user had seen the app in the store and had been motivated enough to download it. A proper understanding of the key reasons behind the frequent uninstallation of new apps is required, so that the mistakes are not repeated.

### **2.4.1. A Complicated App That Takes An Age To Download**

Any app developer worth his/her salt would know that if an app is not user-friendly, it's fate is more or less sealed from the very outset. A painfully slow loading process, complicated in-app navigation, and absence of clear instructions for users can all lead to people getting frustrated with a mobile application and deleting it. A very simple app that is quick to load and easy to use can be a fairly big success, while a large app that crashes frequently is destined to fail.

## 2.4.2. Lack Of Consideration For User Privacy



Mobile app security is something that all app developers around the world take very seriously. Research results have shown that 3 out of every 9 apps pose a threat to user-privacy, and hence are not likely to be retained on devices for long. Online shopping applications, m-payment portals and social networking apps are particularly vulnerable to such threats. In simple mobile storytelling apps for kids too, parents generally wish to make sure that there are no risks of accidental in-app purchases. If security is an issue, the popularity of an app can never soar.

### **2.4.3. Confusing Buttons And Other Tappable Items**

Certain problems can linger in app screens that have apparently neat and organized UI. Mobile app designers might mistakenly (*or in a bid to enhance user-interaction*), include elements that look like buttons, but do not generate any ‘call-to-action’ behaviour when tapped on. Another common mistake is making the actually tappable links and buttons and tabs too small – causing users to face problems while trying to correctly click in the ‘active area’. Things like these irritate users – and from there, uninstalling the app is the natural next step.

### **2.4.4. Apps That Over-promise & Under-deliver**

Users typically download apps by checking out its title, and more importantly, its description in the App Store/Play Store. Now, in a bid for app store optimization (ASO), some developers make the folly of making false claims about the functions of their apps. This strategy might lead to a short-lived initial spurt in downloads, but users would soon find out whether an app actually delivers what is promised in its listing page at the store. If it doesn’t, uninstallation will be prompt, reviews will be poor, and app retention rates will remain shabby.

## 2.4.5. Too Frequent, Irritating Notifications



Sending notifications from time to time is a great way for app developers to remain in touch with users. Some apps tend to overdo this though, sending notifications, emails and other messages too frequently – stuff that most users are not interested in at all. What's more, many of these apps do not provide users an option to 'Turn off notifications'. Hardly anyone likes their phones to constantly buzz with useless notifications – and after a certain level, they can lead to apps getting ousted from devices.

### **2.4.6. App Does Not Function Offline (i.e., no offline mode)**

Online social networking apps, mobile news portals and real-time multiplayer games are all very nice – but an app should have a properly functioning offline mode as well. A large percentage of users interact with apps on their devices at places where network coverage is either weak or totally unavailable (during a trip via underground rail, for example). In case there is no offline mode, that severely hampers the usability of an application – and can prompt users to delete it.

### **2.4.7. Battery Drain & Device Overheating Problems**

A significant percentage of frequently uninstalled apps have a common problem – they cause excessive battery drain. Those who make apps cannot really assume that people will actually ‘kill’ their apps after using them every time (*most users indeed keep apps running in the background*). Certain games have been reported to have caused around 40% battery drain, simply because they were kept running on devices overnight. Device overheating is yet another problem – frequently experienced by users fond of gaming apps. Having to recharge a smartphone after every few hours is not something anyone likes, and for users, it is a

smarter option to identify and get rid of the rogue app.

### **2.4.8. Apps That Are ‘Too Big’**

More than half of the total number of app uninstallations occur due to the concerned application taking up too much of storage space. That's precisely the reason why mobile app companies generally shy away from making apps that are too 'large' in size. The average sizes of an iOS app and an Android app are 6 MB and 23 MB respectively (*the average sizes of games made for the two platforms are 40 MB and 60 MB*). A user might find an app interesting, thereby triggering download behaviour. However, as soon as (s)he finds out that it is hogging a lot of memory space, (s)he is likely to get rid of it and find a 'smaller' alternative.

## **2.5. How To Satisfy App Clients?**

In the previous section, we talked about the probable problems and glitches in a mobile application that can upset individual users. Now, let us focus on the operations of mobile app companies. A person with a valid app idea (the app client) typically partners with a company with the assumption that, the latter will be able to create a proper application - just as it had originally been

conceptualized. Given that the clients often pay big money to app development companies to work on their ideas - they, justifiably, expect uniformly high quality of service (QoS). Matching up to (preferably, exceeding) clients' expectations is an important challenge.

### **2.5.1. Asking For Hefty Advance Payments Should Be A Strict 'No-No'**

Although the overall costs associated with mobile app development are often hefty - that is not an excuse for you to ask for a huge advance payment from your clients. Chalk up a payment schedule at the time of drawing the initial contract, break up your total service charge in 3-4 installments, and mention when the payments have to be made. *25%-30% of the total fee is the maximum you can ask for in advance.* Do not ask for complete payment before the project is complete and the app has been handed over.

### **2.5.2. Providing Cost Estimates Is A Must**

No app client in the world would think about making a mobile application without factoring in budget considerations. If a company does not provide free, detailed app quotes, expect most of its potential clients to move over to another app agency which



does. Uncertainties and vagueness regarding expenses annoy any person, and those who are planning to get an app made are no exceptions.

### **2.5.3. Projects Should Not Be Outsourced To Other Companies**

There are app development companies which are: a) after quick money, or b) looking to take up as many app projects as possible, or c) both. Understandably, they cannot handle all projects on their own, and delegate at least some of their app development work to other, obscure, third-party companies/indie developers. This detail is often kept a secret from the clients. Apart from being grossly unethical (*the client had signed up for service from you, not some other company*), it's a folly to think that people will never find out about this. The difference in quality of the apps will, more often than not, give an indication that they were probably not made in-house.

### **2.5.4. Lags In Query Handling**

Mobile app developers have the responsibility of guiding clients through the entire development process. It is only natural that clients would pose queries (*via Facebook, through email, Skype/phone calls*) – and it is your responsibility to ensure that every question (*even the slightly silly ones!*) are

satisfactorily, and quickly, resolved. Never take more than one day to provide free app quotes, or to respond to any client query. *If your company promises 24×7 service, mean it.*

### **2.5.5. Meeting Deadlines Is Important**

The average client wants apps that deliver optimal value, and are made & released quickly. The tradeoff implied in the '*App Triangle*' is **NOT A LICENSE** for you to dilly-dally on a mobile app development project. You need to have enough qualified manpower to ensure that the absence of a couple of developers can be filled in by others – and the work on any particular app is not interrupted. Complete apps at least a week before the deadline, and keep your clients happy.

### **2.5.6. Violation Of Intellectual Property Rights**

Who '*owns*' the app once it is complete? Is it you, who has coded and designed it, or the client who has paid for it? A client hires mobile app developers and pays them to create applications – nothing more. All the intellectual property rights should remain with him/her, and not the company which had done the coding/designing. Provide a non-competing document/non-disclosure

agreement at the very start, to keep your clients at ease regarding this issue.

### **2.5.7. App Clients Have To Be Kept In The Loop At All Times**

Generally, mobile app developers know all about coding and wireframing and prototype-making, while clients generally do not have any technical knowledge per se. Fair enough, but even so the latter has the *RIGHT* to know how the work on his/her app is progressing (remember, (s)he is the one shelling out the big bucks!). Share mockups, dummy screens, and UI screenshots of the apps to the concerned clients. Make them feel involved in the app-making process. Making a mobile application need not be like a proverbial '*black box*'.

### **2.5.8. Visual Appeal & Ease Of Usage Have To Be Strong Enough**

Taken together, there are over 5 million apps in Google Play Store and Apple iTunes. It's very easy for a boring-looking app to get lost in the crowd. A mobile app agency simply must have a separate team of UI/UX designers, animators, and graphic experts. They are the ones who make an app *LOOK GOOD*, and that is an important factor when it comes to initial downloads of an application. An app client cannot be expected to pay money for a

piece of mobile software that does not even appear attractive.

## **2.6. What It Takes To Make A Successful App**

Every mobile app maker wants his/her application to top the app store download charts. The challenge is not an easy one though - as is evident from the fact that 80% of the available apps in the stores have a total download count of less than 1000, with many even having double-digit download counts. A well-thought-out framework has to be followed, to enhance the probabilities of success of any new mobile application.

### **2.6.1. Apps Should Deliver Value**

There should always be a reason why people should download your application. If an app simply displays information that is already available on mobile-friendly websites, most people will not feel motivated enough to download it. You need to zero in on a unique value proposition for your app (might be entertainment, might be image-editing, might be digital reading, or anything else), and highlight that to prospective end-users. Start doing this before your app is launched. Building curiosity among customers is important.

## **2.6.2. A ‘Simpler’ App Has More Chances Of Being Successful**

Irrespective of the genre of your app, it should have simple UI/UX layouts, a seamless background, and user-friendly controls. Provide an instructions screen to guide the users. People neither have the time, nor the patience, to spend time ‘*learning*’ how an elaborate app works. A complex mobile application (*say, a game with a host of characters, loads of features, and high-end, intricate gameplay*) is not likely to find favor among customers anyway.

## **2.6.3. Educate Users With Tutorial Videos**

Online videos are one of the best ways to do give all the important information about apps to the potential customers. Right from the downloading and installation procedure, to the controls, features and in-app navigation feature – highlight everything in the video. Upload the video on popular channels like YouTube and G+, and share its link on your other social media channels. Explain the user-permissions that your app would seek, and be forthright about the app monetization strategy implemented. The less doubts people have about

an app, the more likely it is that they will check it out.

### **2.6.4. Respect User Privacy**

The domain of mobile app development has evolved tremendously over the last half a decade – and with that has grown the concern about app security and privacy assurance. With mobile payments and personal budget tracking becoming increasingly popular, it is only natural that people would look for apps that rule out unauthorized personal data access in any way. Include an additional screen in your app, where all the '*terms and conditions*' and '*privacy policy clauses*' would be stated. Unless a person is convinced about your app's security features, (s)he will never download it.

### **2.6.5. App Store Optimization**

People come to the online app stores (*iTunes* or *Google Play Store*), search for the type(s) of apps they are looking for, browse through the results, and select one from them. The entire process takes less than three minutes. The trick lies in ensuring that your apps also show up in the customer's' search results. At the time of submission, write out a detailed description of your app. Think up a set of keywords that are likely to feature in people's searches, and include them in the description. For

instance, if you have made a kids' app, specify whether it is a '*mobile game*', or a '*mobile educational app*', or something else.

### **2.6.6. Press Coverage Can Give Your App A Big Boost**

Leading app review sites pick up and feature apps randomly from the store. In addition, there are plenty of free press release distribution sites, where you can publish news, updates and interesting tidbits about your apps. You should also submit your app for evaluation at select free app review sites. Don't lose sight of the app review exchange groups and communities on Facebook and Google Plus. As the buzz about your app will grow, the bigger, high-traffic sites might just feature your app.

### **2.6.7. Mobile App Community On Social Media**

The importance of real-time social media integration in mobile applications can hardly be overemphasized. According to experts from top mobile app companies, social media sharing features are great for bolstering user-engagement, and ultimately, building up a mobile community (*think: a large group of users discussing about data generated from your app*). Make sure that users can share stuff directly from the app to their

personal FB, Twitter, LinkedIn or Google Plus profiles.

### **2.6.8. App Screenshots - The First Point Of Interaction For Users**

At the Apple iTunes store, you have the opportunity of uploading as many as 5 screenshots of your iPhone app. Each screen you choose should display a specific, important feature of the app. If possible, a couple of lines of text should be present on each screen too – explaining the functions of the latter. For Android apps, you will need good-quality app screenshots and an optimized cover image. The first impression of an app needs to be a good one.

### **2.6.9. Pay attention to Design Matters**

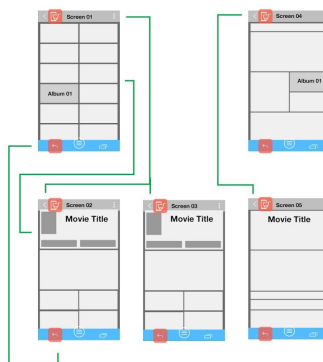
All the images, animations, designs and other visual elements should go with the overall flow of a mobile app. The pictures included in the app display should never distract users and/or disrupt the navigation flow of the application. Similarly, the app development graphics used should never affect its speed.



## 2.6.10. Create Multiple Versions For Cross-Device Usability

Developers need to have separate, customized versions of an app, for smartphones, tablets, and (*if required*) wearables. Research about the popular mobile devices in the market, their screen sizes, display resolution levels, and other such important features. Make sure that your app has custom features – so that the version displayed varies correctly with the device on which it is installed. A mobile app that does not work on tablets (*or vice versa*) has only limited profitability scopes.

## 2.6.11. Smooth In-App Navigation Does Not Force Users To Think Too Much



An average smartphone user is too reluctant to tap their way through multiple screens (*no matter how creatively they are designed*), to arrive at the page they are interested in. Keep the navigation simple, and make sure that all the key features of an app are accessible within a maximum of 3 taps/clicks.

According to a recent survey, 68% respondents stated that long, complicated registration processes led them to get rid of newly downloaded apps from their devices. Make an app that offers practical value and is a joy to use - that's what will translate into its success.

---

# 3. Designing A Mobile App: Graphics & More

---

*In this chapter, you will learn:*

- Key differences between UI and UX
  - Basic app designing guidelines
  - Splash screens and how to create them
- 

In the context of mobile app development, the tasks of coding and graphic designing go hand-in-hand. As per 2017 stats, around 25% of applications are abandoned by users after single-time usage - and one of the biggest reasons for this behaviour is sub-optimal visual features of the software. The graphic elements (images, colours, fonts, videos, animations), along with the overall in-app navigation scheme, play an integral role in the success - or otherwise - of a mobile app. Let us put focus on that in this section.

## 3.1. The Differences Between UI Designer And UX Designer

In common parlance, the terms 'UI' (*user-interface*) and 'UX' (*user end-experience*) are often used synonymously. However, using the two terms interchangeably is erroneous, for, while the roles are broadly similar (*as part of design processes*) – there are important inherent differences between the two. Before diving into the do's and don't-s of app designing, let us first understand the differences between these two concepts.

### 3.1.1. Design Principles

A UI designer is typically concerned with the visual appeal of the product (*say, a mobile application*). (S)he combines different forms of typography and colors to satisfy the precise requirements of clients (*note: the final user is NOT the chief point of concern here*). A UX designer typically is more interested in human-centered design principles – how the implemented design impacts the interactions with the product. Instead of fonts and colors, (s)he is more involved in broader aspects, like task flows and simulated environments/scenarios.

### **3.1.2. What is A ‘Good UI’ and what is a ‘Good UX’?**

If a website or a mobile app is beautifully designed (*very creative splash screens, great home page/screen et al.*), that’s a great example of a ‘good UI’. However, that does not automatically imply that the said site or app has a ‘good UX’ as well. For that to happen, the usability of the site/screen has to be optimized, so that users get the best possible experience from accessing it. By the same token, an app with a ‘good UX’ in theory might have a ‘horrible UI’ (*think about elaborate flowcharts and design plans falling in the hands of inept designers*). Both UI and UX are necessary, and neither of them are sufficient without the other.

### **3.1.3. What Do UI Designers and UX Designers Deliver?**

Everyone involved in a mobile app development project has deliverables. UI designers and UX designers are no exceptions either. Project sitemaps, prototypes and high-fidelity mockups, storyboards – all of these fall under the deliverables of a UX personnel. The UI designer works on a more micro level, and is generally responsible for delivering all the individual visual elements, as well as defining the behaviour/flow between them.

### 3.1.4. Scope of operations

Purely from the respective scopes of operation of UI designers and UX designers, the latter cover a much wider area than the former. UX covers the overall user-experience with any product, how people interact with it, and how their needs are being fulfilled (*and obviously, UX need not be necessarily associated with tech products*). UI, on the other hand, works on a more ‘micro’ level – dealing with the actual appearance and feel of the product, the buttons, tabs and clickable areas, and all the other elements that make up the overall user-experience. In a nutshell, UX is about optimizing the user-experience, and UI is about how it is done, on a granular level.

### 3.1.5. Responsibilities

Designers, whether dealing with the UI or the UX of any product, have to be creative – there are no two ways about it. However, UI designers also need to be masters at ‘convergent thinking’, for identifying and deciding the ideal design elements/interactions to be used on a product. For UX designers, ‘critical thinking’ abilities are more important. With the product at the center, they have to zero in on a uniform, optimal, predictable behaviour of the user. A UX designer simply has to be present in a project

right through, while the task of a UI designer is limited to a stage.

### **3.1.6. UX is a broader concept**

UX-designing is an exhaustive, all-encompassing job – and UI is present as a subset within it. Also included in the domain of User End experience are information architecture, audio, video and text content, interaction designs (*user-product interactions*) and industrial designs, in addition to visual designing. UX can be summarized as ‘Designing For Emotion’ (*as done by Mailchimp’s lead UX designer*) – in the sense that, it ‘*influences*’ the emotions and ‘*conveys*’ a message to the user, when (s)he uses the concerned product. UI designing is one of the many design responsibilities under UX.

### **3.1.7. UI is tangible**

Every element in the user interface of a mobile app can be seen and touched. These are the tools with which a user interacts with the application, the way in which (s)he can control it. UX is, however, intangible – and deals with the overall experience (*good or otherwise*) that a person has while using the app. For instance, if a button on the mobile screen is easily visible and tappable, that amounts

to a good UI – but if it takes a long time for the next screen to load, the user-experience is ruined.

### **3.1.8. Tools & Technologies Used**

Given the basic similarities in the roles of UI and UX designers, it is not surprising to note that the set of tools/software tools they work with have a lot in common. For instance, both are likely to be proficient in using Adobe Illustrator, Adobe PhotoShop and Sketch. However, tools for mobile/web prototyping (InVision, for example) are exclusively in the domain of UX designers. They frame out the overall behaviour flow for a product, while the onus is on UI designers to maintain the consistency in the visual designs, and ensure that the plans laid out by the UX professional are properly implemented.

## **3.2. How To Design A Mobile App**

Brilliant app concepts count for very little if graphic designers botch up the interface of the application. If you browse through the 5 million+ apps at iTunes and Play Store, you will come across many applications which are difficult to even understand, let alone using them. A simple rule of thumb needs to be followed – mobile app designing should always be about facilitating app-usage, and never about the designers/animators showing off their



professional skills. We will discuss some important guidelines for mobile app designing here.

### **3.2.1. Choose the ‘design scheme’ with care**

*‘It looks good’* is never a good enough reason for choosing any particular layout/design theme for an app. Most mobile app development companies, while providing online free app quotes, request users to share preliminary layouts, and provide a brief summary of what the application would be all about. Use this data while designing the concerned app (*i.e., the mobile app design chosen should be relevant to its genre/type/specific features*). And yes, if that means sacrificing some of your creativity, so be it.

### **3.2.2. Put user-behaviour in focus**

There are two ways of conceptualizing how the screens/interface of an app should be designed. The first is the one which would be easy for the UI/UX designers at your mobile app company, while the other is the one which focuses on user-convenience. The latter should be the one you focus on. Remember, the success of an application is not determined by how easily it can be designed, but whether it is finally deemed to be user-friendly (*easy-to-understand menu bars, smooth in-app*

*page navigation, etc.*). Put ‘designer needs’ in the back-burner, and let ‘user needs’ take centerstage.

### **3.2.3. Device-compatibility is vital**

Gone are the days when you could create an iPhone app and be done with it. Given the overwhelmingly larger market share that Android enjoys, cross-platform mobile app development and designing are skills you need to muster. In addition, keep track of all the new and popular mobile devices on which the app would be used, and optimize the design layouts accordingly. The last thing you want is anyone complaining that your new app is not properly viewable on his/her device.

iOS app developers have to create separate, customized app versions for the iPhone, the iPad, the iPod Touch, and (if required) the Apple Watch and Apple TV.

### **3.2.4. Use icons wisely**

Contrary to what many believe, the first point-of-interaction (PoI) between the user and a mobile application does not happen AFTER the former has downloaded the app on his/her device. This first-time interaction takes place right when the user sees the app icon for the first time in the store. Therein lies the importance of coming up with a

good app icon - to make that all-important first impression on people, catching their attention and (*hopefully*) triggering download behaviour. The app icon should provide an idea about what the application is all about.

### **3.2.4.1. Icons in app screens**

New developers make the mistake of pushing in as many words as possible on the, relatively speaking, small app screens. Using icons in place of words would be a much smarter strategy. For starters, you would save a significant amount of space (*a nice and clear icon would take up a lot less space than even a 3-letter word*) – and as we know, keeping app screens uncluttered is of essence. What's more, app icons, over time, help in marketing/branding purposes. When chosen and used properly, the icon can become the 'symbol' of a mobile application...at times a repeated icon can be more identifiable to users than even an app's name.

### **3.2.5. The importance of reiterations**

Contrary to what many mobile app developers believe, making apps is NOT a one-shot game. You need to implement the preliminary design elements, test the app, make changes (*if necessary*), iterate the entire app testing procedure, and so on (*this*

*chain should continue until all design flaws have been ironed out*). Do not rely only on automated testing, and get feedback from human testers (*ideally, form a focus group*). The more the number of app reiterations you do, the less you would have to worry during the final mobile app testing phase.

### **3.2.6. Keep space for virtual keyboards**

An otherwise good-looking app screen can appear messy and cluttered as soon as users call up the virtual keyboard (*e.g., to provide text inputs*) on it. In fact, this is often one of the reasons why many promising new mobile apps for kids fail every quarter. During the app testing phase, you need to check every screen of the app, with the keyboard displayed on it. Do not just assume anything about user-behavior, you are more than likely to be proved wrong.

### **3.2.7. Seamless in-app navigation**

Unless an app is user-friendly, it is of no value. Make sure that the overall onboarding process is easy, and the in-app navigation is uniformly smooth. All tappable areas/sections should be clearly highlighted - and, as a rule of thumb, a person should not have to tap/swipe more than 3 times to reach the screen (s)he is interested in. In

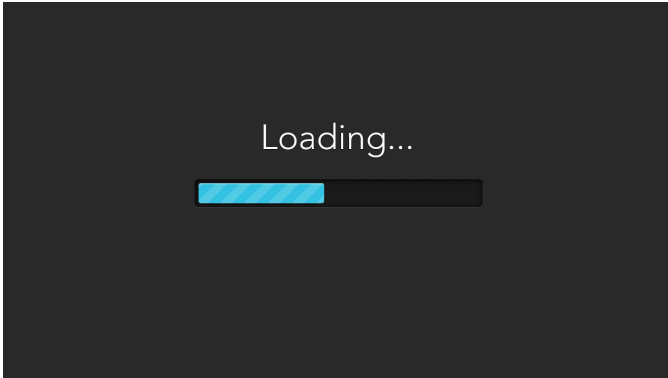
kids' applications, the controls have to be child-friendly.

### **3.2.8. Make use of existing resources**

Thanks to the emergence of so many online mobile app development forums and communities, there is no paucity of app design tutorials and case studies. Make use of such open source resources as much as possible (without, of course, stooping to the level of full-on copying!). Share your app design ideas with peers, and learn from their successful (*and failed*) projects. While designing an Android or iOS app, you might overlook a design mistake – but the same can get easily spotted if there are other professionals taking a look at your work.

Mobile app designing is a specialized task. It would be a folly to assign design-related tasks to the developers. In general, coders do not make good graphic designers - while creative professionals are not programmers. There should be separate teams for designing and development - closely collaborating while working on an app project.

### 3.3. The ‘Splash Screen’ Factor



Ever noted the ‘loading’ screen that appears when you launch an app? That is called the ‘splash screen’ - the screen that is displayed as all the elements/assets of the app gets ready. The splash screen - once again - is one of the first points-of-interactions (Pols) of an individual with a mobile app, and designers need to prepare it very carefully.

#### 3.3.1. An overlong splash screen is a bad splash screen

A splash screen display should never go on beyond a maximum of ten seconds (*if anything, it should last between 5-8 seconds*). As soon as the app has loaded, users should be redirected to its main page. On average, 1 out of every 4 mobile users

close/uninstall apps simply because they have too long splash displays. A quick, short intro – that's what you should create.

### **3.3.2. Include dynamic elements (e.g., loading progress)**

For all your creativity and imagination used on a splash screen, the latter is nothing more than one (*or, a series of*) still images. There's no scope of user-interaction on the screen – and many new mobile users might not have the patience for the entire splash display, even it lasts for only a few seconds. Put in a display bar that shows the loading progress of the app. It will lend a dynamic feel to the screen, and would assure users that the main app would be launched soon.

### **3.3.3. Avoid using heavy graphics**

There is no point in writing long lines of text on the splash screen – simply because no one would get the time to read them. Ideally, use an interesting tagline, which offers viewers a proper initial idea about the application. Use simple yet elegant app development graphic themes on the splash screen. Otherwise, the screen might become slow, hurting the overall performance of the app.

### 3.3.4. Keep room for the ‘bleed area’

Since the aspect ratios of mobile devices vary from one to the next, this is something you need to consider. Any app designer worth his/her salt would advise to keep around 100px of ‘*bleed area*’ on the left and right side of the screen, and approximately 200px at the top and at the bottom. Do not put anything inside these ‘*bleed*’ margins, they are not likely to be visible on most devices.

### 3.3.5. Alternative resolution levels

What looks great on an iPhone 5 handset might look distorted on the iPad – and would, in all likelihood, appear too stretched on the iPhone 6 (*which has a larger screen size*). In general too, the average resolution levels on the display of Apple phones is radically different from that on Android handsets (*e.g., the Samsung Galaxy range of phones*). While creating separate splash screens for every model would be way too time-consuming, you should ideally have three screen versions ready. That way, you can implement low, medium and high-resolution screens, depending on the device an app is downloaded on.



### **3.3.6. Put creative work in the center**

In spite of careful cropping and resizing, the edges of a splash screen might not be visible/appear blurred on certain smartphones. If you have images/text in these areas, they would not be viewable. You can tackle such probable problems, by putting in all your designs and other UI/UX creative work at and around the central portion of the screen. No one should have a problem in getting a clear view!

### **3.3.7. Splash screens can be great branding tools**

Considering a mobile splash screen to be only a medium to show off cool animation effects would be rather naive. You should ideally put your company as well as the unique app logo on it, along with other visual branding elements (images, symbols, punchlines, etc.). The screen would be displayed every time the app is launched, and you can easily use the former to enhance the general brand awareness levels.

### **3.3.8. Zero disruption to app usability**

A simple, elegant splash screen always works better than one overcrowded with pictures and/or is too slow. Avoid using any element in the screen

that might distract viewers, and make the screen optimized for devices that have relatively low-speed internet connectivity. Tasteful images, shown in an interesting manner before the main screen comes up, need to be used. Make sure that the size of their APKs is not increasing too much due to the inclusion of splash screen patches.

A well-designed mobile app splash screen needs to be properly scalable - to suit the varying display resolutions of the different compatible devices. It should not linger on for long - and for the seconds it remains visible, it should do a good job of keeping users interested.

---

# 4. Storyboarding And App Testing

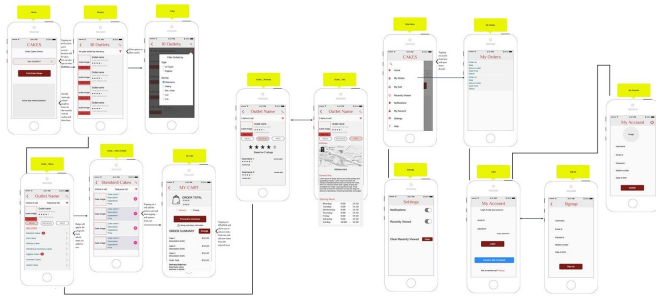
---

***In this chapter, you will learn:***

- Wireframes
  - The importance of app storyboarding
  - Types of mobile app testing
  - Things to test in an app
- 

Now that we have got the basic app development and designing concepts sorted out, let us turn our attentions to a couple of slightly more technical topics related to app-making. First, we will very briefly talk about the importance of wireframing. The advantages of the storyboarding (*available for iOS developers since the arrival of iOS 5*) will be highlighted next. Then, we will move on to the all-important topic of mobile app testing.

## 4.1 Wireframing

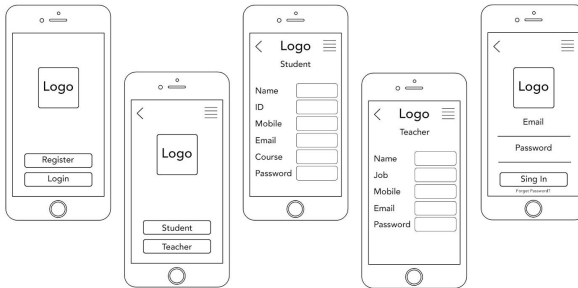


Before builders start out with a new construction project, they typically prepare a detailed blueprint. Similarly, developers also need to prepare a skeletal, low-fidelity, structure of new applications (*in essence, visual representations of app ideas*) - before starting the coding process. These preliminary sketches are called ‘*wireframes*’ - and they are typically used to understand what each screen of an app is supposed to do. With wireframes, developers can easily define the foundation, structure and hierarchy of applications.

Given the importance of wireframing, the availability of many high-quality wireframing tools - each with different features and capabilities - is a definite advantage. Some popular app ideation/wireframing tools are Moqups, InDesign CC, Axure, FluidUI, Visio and UXPin.

## 4.2. Why is Storyboarding important for iOS developers?

### APP STORYBOARD



Storyboards are handy visual interface builders available for developers working on the iOS platform. They serve as graphic organizers, with screens and illustrations and animations placed sequentially - for a better, more in-depth understanding or a starting point for app developers. In this section, we shed light on the main advantages of using the storyboarding technique.

### 4.2.1. Self-explanatory visual representations

Apart from offering real-time app design and transition views, Storyboards come in real handy for table-viewing. The feature has built-in static cells and prototype cells functionality – which helps iPhone app developers to iron out errors while working with tables in the app infrastructure. Doing the same via manual coding involves a lot of extra hassles.

#### **4.2.2. Adding data and code snippets is easy**

Many iOS app developers (particularly those who have taken the pain of having to manage delegate patterns earlier) love the segue unwinding option in Storyboards. With segue, relatively short and simple method calls can be used to build up the interaction between the view controllers that are being used. As already pointed out, with XCode becoming progressively more robust, chances of errors while adding data/code snippets are also going down.

#### **4.2.3. Scene transition ease**

The chief purpose of Storyboards has always been facilitating scene transitions in apps – and it's fair to say that they more than satisfactorily fulfill this requirement. Developers can use several types of Segues (depending on the precise need of their projects) – ranging right from Modal and Push Segues, to the Unwind and the Sourceless Segue options. There is even a Custom Segue feature – which can be maintained as a subclass of *UIStoryboardSegue*. To make app UIs properly customized, Storyboards are indeed a big help.

#### **4.2.4. Creating app prototypes and ensuring compatibility with multiple devices**

Creating detailed app prototypes is vital, and Storyboards are instrumental for this purpose. Most contemporary companies are into making apps that are compatible with a wide range of iOS devices. This is yet another area where storyboarding proves to be a smart option. Multiple views can be organized with ease on a Storyboard. The alternative would have been separately naming the files – and that is not a convenient option. At all.

#### **4.2.5. Creating high-fidelity mockups**

Storyboarding allows users to create mockups of their apps, without having to write big programs and complicated codes. As a direct result, chances of app designing mistakes are also lowered. For creating mockups of certain types of app designs with Storyboards, there is no need for coding at all. Fair to say, storyboarding has taken away much of the complicacies associated with creating mockups.

#### **4.2.6. Making a ‘demo’**

Nearly all decent mobile app companies have the policy of sharing demo views of apps to clients. Working with Storyboards is the simplest possible way to add mock data to the user-interface of any project. With repeated additions, the overall workflow/functions of the app can be built up – and that too, with minimal amounts of coding. The demos created with Storyboards are generally interactive, which makes it easy for clients to check their projects thoroughly, and give their feedback.

#### **4.2.7. Use when required**

There is no compulsion that Storyboards have to be used right from the start of any iOS application development project. A developer can simply uncheck the Storyboard option when (s)he starts to make an app – and move over to storyboarding later. Starting off new projects with Storyboards is



often unnecessary, and it's easy to avail of this feature whenever it is actually required.

#### **4.2.8. Tackling design iteration related problems**

The most proficient of app developers would agree that programmatic UI/UX designing involves multiple iterations (which can be troublesome) – at least in the initial stages. Storyboarding is probably the best way to do away with such problems. Correct (read: *'pixel-perfect'*) design placements at the first go becomes easier than ever – and what's more, making changes is a straightforward process too. On this count, Storyboards are way more flexible than XIB.

While storyboards are not suited to multi-user management (files cannot be changed by more than one user at a time) - they offer significant advantages over programmatic development. With the help of efficient storyboarding, developers can actually speed up app development processes. In addition, working with storyboards also help app-makers to move beyond the 'coding level', and view the project from the 'presentation level'.

### **4.3. Mobile App Testing**



Buggy apps are bad news. Users have little tolerance for them - something borne out by the stat that only 15% people can actually be bothered to try an app (*that had not worked as expected the first time*) more than twice. With no shortage of similar, substitutable apps in practically every category - this brings to light the importance of thorough app testing. We will elaborate on this aspect over here.

### 4.3.1. The Types Of Testing

A one-dimensional testing method is not good enough, while checking the quality of service (QoS) of a new mobile application. Bugs and errors can crop up from various fronts - and if they remain undetected, they can cause serious damage to the end-experience of users. Here are the types of

testing that professional app testers need to perform:

#### **4.3.1.1. Performance Testing**

A potential source of confusion – since by ‘*performance*’, many new app developers understand the user-experience that an app is likely to deliver. In truth though, the scope of ‘performance testing’ is a lot wider than that. It includes checking whether there are any bugs/problems in the client-side architecture, server-side setup and the network functionality of the application. In summary, all forms of front-end testing of an app come under performance tests.

#### **4.3.1.2. Functional Testing**

Mobile app developers require functional testing to find out whether there is any problem with the included app content (ranging from text and links, to images and videos). Detailed functional testing is also important for monitoring app analytics later on.

#### **4.3.1.3. Load Testing**

It might very well be possible that a mobile app is being simultaneously used by a large number of people. Thorough testing is done, via separate metrics, to find out how the response times,

functionality, server load, and other features are likely to be affected in such cases. Multiple test scenarios are created, to compare expected with actual usage levels of an app.

#### **4.3.1.4. Interface Testing**

For this, proper collaboration between the coders, UI/UX designers and the app testing department is essential. The in-app navigation flow, including the arrangement of menu buttons and other tabs, is checked at this stage. Most developers follow a thumb rule that final users should not have to tap/click more than 3 times to reach the app screen they are searching for.

#### **4.3.1.5. Operational Testing**

What if an app has good enough features – but causes loss of unsaved data in case the phone battery runs out when it is being used? Or what if there is a screen freeze? To counter the potential risk of loss of app data, operational testing is essential. The overall data backup and recovery process of a mobile app is examined at this stage.

#### **4.3.1.6. Device Testing (or, Compatibility Testing)**

When a developer creates an iOS app, (s)he has to check whether the app is compatible with the latest flagship iPhones (7 and 7 Plus), as well as iPhone 6/6S and iPhone 5 (*decisions on further support for older models have to be taken by individual mobile app entrepreneurs*). Most iOS apps have to be compatible with iPad and iPod Touch as well. For Android apps, device interoperability testing is even more challenging, due to the sheer range of new Android devices made available by different vendors every year. A list of all the popular handsets has to be drawn up, and the app has to be tested on each of them. During the development stage too, testing the codes across multiple browsers is advisable.

#### **4.3.1.7. Interruption Testing**

Any decent app developer would know that phone calls/text messages/emails can, and probably will, ‘interrupt’ users as they are checking out a newly installed app. Hence, it only makes sense that a portion of mobile app testing is dedicated to find out how the app handles such ‘interruptions’. Interruption testing has to be done via emulators as well as on actual devices.

#### **4.3.1.8. Memory Testing**

Since mobile devices typically have a fraction of memory resources compared to desktop systems, apps that hog too much of memory and consume excessive bandwidth are likely to be slow, and have a propensity to crash often. For iPhone app development experts, memory testing can be done by installing the app on a device, and connecting it to a compatible Macbook. Testing the memory usage of Android and Blackberry apps is fairly easy too.

#### **4.3.1.9. Security Testing**

With the rise of m-commerce apps and finance-related enterprise applications, the importance of security testing has also spiralled. In order for a business app to be successful, it should boast of foolproof data security and integrity features. The data stored in the app (*or in associated hybrid cloud*) should never be accessible to any unauthorized individual/organization.

### **4.4. What To Test In A Mobile App?**

On average, smartphone users wait for only 3 minutes to check whether a newly installed application loads/functions correctly. If anything goes awry, the app at fault gets uninstalled - and

another, similar app is found and downloaded. App developers have to be proactive and put themselves in the shoes of the final users - while determining the features to test in a new mobile app:

#### **4.4.1. Testing the app interface**

The success (*or otherwise*) of an app hinges crucially on how easy (*or otherwise*) it is to use. There is a reason why app development companies have separate teams of graphic designers and animators and UI/UX designers to manage the visual appearance as well as the in-app navigation of mobile applications. Prior to release, it has to be tested whether all the menus, tabs, bookmarking options and other settings are working properly. An app may have powerful features – but unless users find it easy to use, popularity levels will remain low.

#### **4.4.2. Backward compatibility**

Mobile app development experts have to take a call on the degree of backward compatibility (*i.e., support for older versions of the mobile platforms*) that they should add to their applications. Most iPhone applications reach back to iOS 7 in terms of compatibility, while some are even supported on older, iOS 4 handsets (*there is no definite pattern for this on Android, although the oldest supported*

*version for most apps is Android 2.2 Froyo).* Creating an app only for iOS 10 or Android 7.0 is not advisable – since that inevitably limits its overall reach.

### **4.4.3. Reliance on network speeds**

There is simply no way to correctly guesstimate what the network connection speed of the users' devices would be. There will be people using apps on Wifi-activated devices, some will be using 3G mobile data – while there will be a fair number of 2G (*and there is the EDGE/GPRS/CDMA fragmentation here*) users as well. The performance of an app should not be affected by the network speed of the devices it is installed in. What's more, most apps and games should be usable in offline mode as well. It's all about maintaining high app-engagement levels.

### **4.4.4. Download, installation and login**

Let's put it this way – why would anyone bother wasting time on an app that cannot be downloaded at one go from the app store? After a quick download, users should also be able to easily sign in/sign up on the applications. Testers have to install/uninstall and create accounts of applications on devices (*emulated devices/simulators won't do*



*the trick here*), and make sure that there are no glitches on this count.

#### **4.4.5. Privacy**

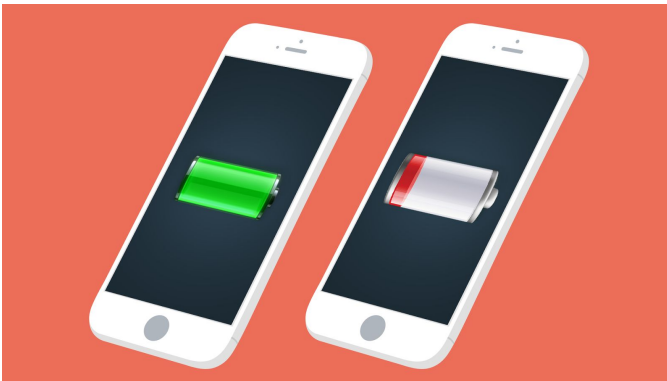
Any Android or iPhone app development expert who knows his/her job would be aware of the growing importance of mobile app security. With platforms like Apple Pay and Android Pay growing in adoption rates, users are understandably sceptical of using any app for financial transactions that do not provide complete security and data confidentiality/protection assurance. Security is also a big factor for all the new-fangled social networking apps mobile apps for kids that are released every quarter. For apps that support m-payments, speed is also a critical issue.

#### **4.4.6. Checking the form factor**

Platform and device fragmentation aside, form factor is yet another important (*and curiously, often-neglected*) aspect of testing mobile applications. Many new app developers make the folly of releasing the same version of an app for smartphones and tablets – and unsurprisingly, the results are disastrous. When the same app is targeting different devices in the same platform (*iOS or Android*), developers have to create separate form factors of the app, to ensure

error-free viewability on different devices. Once again, the growing popularity of wearables have added an extra dimension to form factor testing for apps. The more smart devices vary, the greater is the challenge.

#### 4.4.7. Effect on device battery



Neither Apple nor Google has yet been able to really crack the battery-performance puzzle of smartphones. On top of that, if an app causes excessive battery drainage, its developers can pretty much kiss the chances of its being successful goodbye. It is the responsibility of mobile app testers to make sure that installing and using an app does not put any extra pressure on the battery of a device. An application needs to be tested in high, low and critical battery conditions, as well as when the handset is being charged (*certain*

*apps tend to eat up more battery during charging*). Since most average users do not bother actually closing apps after usage, testers also need to check how device battery is affected when apps remain running in the background.

#### **4.4.8. Interruptions**

Developers should never make the mistake of thinking that using their apps would always be an interruption-free experience for users. While working on an application (*or playing a game, for that matter*), a call can come in, a text message can be received and other notifications can be generated on the device. It is important to note how the app that is being tested behaves when such '*INTERRUPTIONS*' happen. These interruptions can also happen when an app is being downloaded or upgraded. Ideally, the app should work parallelly/get automatically paused during interruptions.

Coders and designers have to put in the hard yards to come up with a nice, efficient mobile app. Then, the onus shifts on to app testers - who have to ensure that this hard work becomes worth it.

---

## 5. User Experience: The 15-Seconds-App-Rule

---

*In this chapter, you will learn:*

- Elements of the '15-Seconds-App-Rule' to optimize FTUE (first-time-user-experience)
- 

How long does a new app have to impress users...so that it is actually 'retained' on devices after being downloaded? Reports have shown that, 8 out of every 10 apps are uninstalled soon after download - indicating the presence of problems. This, in turn, brings to light the importance of the '15-Seconds-App-Rule'. It is all about making a great early impression on final users - and this rule highlights how you and your mobile application can achieve that. We will here learn about the main components of this rule of thumb for developers.



## **5.1. Download and device compatibility**

High-speed internet connectivity is common in even the relatively lower-end smartphones. It should not take more than a 5-7 seconds for a new app to be downloaded. If the download keeps getting interrupted repeatedly, do not expect users to linger around (way too many alternatives are available). In addition, your app needs to have custom versions for smartphone, tablets, phablets, and (if applicable) smartwatch/smart TV. In case your app is meant to be used only on mobiles, specify that clearly in the store description. There should never be any ambiguity.

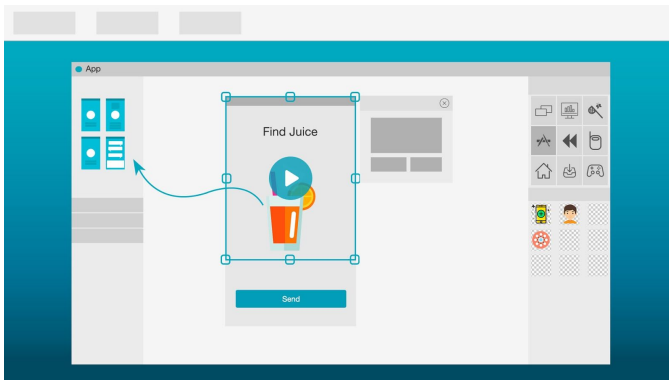
## **5.2. Ensuring good reviews and ratings at store**

Word-of-mouth publicity plays a big role in determining whether a person ends up downloading an app or not. If your app has any sort of performance glitch, expect 1\* ratings and negative reviews to pile up at the app stores. While browsing for apps at the Google Play Store or Apple iTunes, people would, understandably, give such poorly-reviewed applications a cold shoulder (even if you release bug-fix updates soon, the damage is already done). Make sure that the first point of interaction (POI) between your app and its prospective users is a positive one. Focus on quality at all times, so that negative reviews are always at an arm's length.

### 5.3. The app has to load quickly

The generation-Z smartphone owners have thin patience levels – and they do not have the time for apps which take a long time (anything more than 8-10 seconds) to load. The splash screen, which is displayed as an app gets ready for use (i.e., loads) after a user has tapped the app icon, should not be visible for more than 2-4 seconds. There should also be something dynamic about the splash screen (say, a progress bar) to keep the attention of users from wavering.

### 5.4. The importance of ‘Empty State Designing’



How do you judge the pickup of a motorcycle? You find out how long it takes for the bike to go from 0 to 60 mph, right? Drawing the same analogy for a

new mobile app, you can broadly divide its presence in 2 two different states. First is the 'empty state', when a user has only just downloaded the app – and second, the '60 state', when users have already been using (and have grown familiar with) the app. The '15-Seconds-App-Rule' strongly recommends app designers to prepare interfaces for the 'empty state' – where there is no recorded data to be displayed. Unless users are impressed by the look and feel of the 'empty state', why would they be bothered about staying on till the '60 state'?

## **5.5. Pre-registration tour of app**

Do not just push the registration screen as soon as the app is launched by new users. Let users swipe through the app screens as 'Guest', get actually convinced about the value proposition of your application – and then prompt them to sign up. The onus is on mobile app developers to determine the level of this free (pre-registration) access. A screening questionnaire can be added to gather an idea of the users' tastes and preferences. Of course, there should be multiple sign-in options – through email/password, and social media (Twitter, Facebook, (if applicable) Instagram). According to survey reports, an app can lose up to 56% of its user-base, if registration is mandatory BEFORE people can check it out first.



## **5.6. Quick and easy onboarding**

Think of ‘onboarding’ as holding your user’s hand and showing him/her the way around your app. It is generally not required if your application does not require any user inputs. Even in cases where onboarding is necessary, ensure that the process is fast – and only the main features and controls are showcased. Avoid making an elaborate tutorial involving all screens and every feature – since that would only bore people. Instead, provide a short guide, and let users learn the rest by interacting with the app. There should be an option to ‘skip’ the onboarding/tutorial as well. After all, a person downloads a mobile app to USE it – and not spend time learning about it.

## **5.7. Usability in different scenarios**

Let’s talk about an average smartphone user. While on the bus, he can play a mobile game; before going to sleep, he can fire up a meditation app; and while on the move – he can check out news aggregator applications. Prepare your application in such a way that it can be used, whenever, wherever, and for as long as the user wants. Most apps and games should have offline modes as well – so that they can be used even when there is no/poor network coverage. Pay attention to the

screen orientation of your app as well. Users should be able to toggle between portrait mode to landscape mode with ease.

## **5.8. An appropriate icon for the app**

The logo, icons and screenshots you choose for your application are extremely critical. Prospective users should get a fair idea of the nature of a new mobile app by simply checking out its logo and icon...even before they move on to the app description section. Upload the most relevant screenshots of your app for display at the stores. Double-check to ensure that there are no errors in these screenshots. The '15-Seconds-App-Rule' does not start AFTER an app is downloaded – it starts from the moment a user arrives on your app's page at the online stores.

## **5.9. Users should feel that they are in control**

While using a mobile app, people like to be in control at all times. The controls and overall in-app navigation should be very easy-to-understand and execute. The 'call-to-action' points (e.g., 'Add to cart' in a mobile shopping app) should be clearly visible, easily tappable, and completely distraction-free. As a rule of thumb, it should not take more than three screen taps for a user to find

whatever (s)he is looking for in an application. If you are working on a mobile app for kids (a storytelling application, for instance), be extra careful about including child-friendly controls. At every point, let your users know what they can/should do next. Without this prompting, people might feel confused and drop-off from your app.

## **5.10. Presence of a ‘WOW-factor’**

Unless you are making a ground-breaking app (something too innovative for its own good!) – you need to include something extra...something over and above the core feature of the application. That would serve as the unique ‘wow factor’ of your app – adding an extra layer to the attractions of your application. For instance, offer free in-app currency (coins, stars, points, credits, etc.) for the first few hours after an app is downloaded. You can also encourage people to invite their social media friends to the app – in exchange of rewards (like, say, in Uber, where you can get handsome discounts on cab fare when you send out invites and friends sign up). If your audience ‘likes’ your app, that’s no longer enough. They have to ‘love’ it.

The ‘15-Seconds-App-Rule’ focuses on enriching the ‘First Time User Experience’ (FTUE) associated

with a mobile. Things like ad quality and social integration are also important components of this rule.

For having any chances of success, an app has to ace the '15-Seconds-App-Rule'. Making users like an app from the very outset is something that developers should always be concentrating on.

# 6. Mobile Game Development: Tips & Tools

---

***In this chapter, you will learn:***

- Pointers to develop mobile games.
- An overview of the leading game engines and tools.

---

Games make up nearly 25% of all the apps listed at the Apple App Store. In May 2016, as many as 20958 new games were submitted by developers - making it by far the biggest category in the App Store (educational apps, with <2500 new submissions, were a distant second). According to official estimates, the total number of apps in the App Store will reach 5.06 million by 2020 - and gaming applications will be a key driver of this growth.

We will start out this chapter with some basic tips and guidelines for making a mobile game. Game

development engines and other related tools will be discussed next.



## **6.1. How To Create A Game App?**

Mobile games are, and will always be, insanely popular. The available variety of games is always expanding - and thanks to the recent advancements in the domains of virtual reality (VR), augmented reality (AR) and artificial intelligence (AI), developers have so much more room for innovation than even a couple of years before.

### **6.1.1. The starting point should be a 2D game for iOS/Android**

As a beginner, you should start out with a relatively simple 2D game. For creating the game elements and assets, you will need to have in-depth

knowledge of working with Photoshop. There are several 2D game engine tools available as well (e.g., GameSalad), which you can refer to – in case your programming expertise is on the lower side. With time, start learning 3ds Max, Maya and similar applications, which would help you graduate to 3D games. To ensure high visibility of your game, choose between the Apple iOS and the Google Android platforms. Getting a new app approved on the latter is easier, but app and game developers typically make more money from Apple App Store.

### **6.1.2. Pay attention to game content**

Many developers make the mistake of focusing too much on the graphic designs of games – while the actual game content takes a backseat. This should never be the case. The layout, display, controls and UI/UX of a game should complement a strong, engaging, unique content. No one downloads games to check out how beautiful it looks – it's all about whether the central theme of the game appeals to them.

### **6.1.3. Keep things easy for users**

The interaction time of users with gaming apps is low – generally, a few minutes at a stretch. If you put in a host of never-seen-before features in your game, do not expect people to actually spend time

to ‘*learn*’ how it works. Keep the gameplay options simple – so that users from any age group can try their hands on it, without having to refer to the instruction manual. The moment your mobile game is perceived to be ‘*difficult*’, its download figures will plummet.

#### **6.1.4. Choose the game genre carefully**

Selecting the genre for a new game need not be as difficult as it is often made out to be. Any good Android/iPhone game developer would agree that the best idea is to do a research on which genre(s) are the most popular at present, and make your game belong to it. Arcade games are in vogue now, and action-based games (like treasure hunts) have plenty of takers too. Avoid making a game in a genre that is not generally liked.

#### **6.1.5. Identify the target audience**

Are you making a mobile game for small kids? Will your game mostly be played by adults on the go (think about those people furiously tapping away at their smartphones while traveling!)? What would be the income bracket of the people who are most



likely to download your game? These basic queries need to be resolved, before you get down to make an Android or iPhone game. Features like in-app purchases, inclusion of violence in the gameplay (for combat games) should be decided on the basis of the profile of your target audience. In a fun game for kids, you will need to include child-friendly controls too.

### **6.1.6. Motivate users to stay engaged**

There are two basic thumb-rules that any mobile game development expert must follow: Firstly, the game should never ‘*end*’, and there should be many levels which the users can gradually progress too. Secondly, and perhaps more importantly, there needs to be missions/objectives of the gameplay. A user should feel that the entire game has a goal (maybe collecting coins, getting reward points, unlocking secret treasures, etc.). A purposeless mobile game would soon bore people.

### **6.1.7. Give your game a great USP**

It is vital to differentiate your game from the thousands of others already present at the online app stores. Pick a feature of your game that, while not necessarily unique, can serve as its main USP (for instance, one-shot death, star system,

online/offline modes, multiplayer options, etc.). Make sure that all your hard work does not lead to the creation of *'just another game with nothing special about it'*.

### **6.1.8. Take help from resources and tutorials**

From sources like Lynda, Cartoon Smart and Digital Tutors (to name a few), you will find a lot of helpful pointers for making your very first gaming app. Network with other mobile app and game developers, and find out what other resources/references/game samples they use as reference. The more you know about making a game, the better it will turn out to be.

## **6.2. Game Development Engines**

Professional game developers typically make use of multi-featured software frameworks to create 2D/3D games (for mobiles, desktops, consoles, etc.). These frameworks are known as 'Game Development Engines' - and they are the most important tools to be used by developers. In this section, we will take a look at some popular game engines:

### 6.2.1. Unity 2D/3D



As many as 21 different platforms are supported by the Unity game engine – and it is easily among the most-loved tools for creating custom mobile games. With the release of Unity 5 (*the latest update is v.5.6.0*), a wide range of new features have been provided to users. From custom graphics interfaces and excellent rendering support, to the well-stocked Unity Asset Store – this game engine has it all.

## 6.2.2. Unreal Engine



When Epic Games initially started out with the Unreal Engine (UE) game engine, it was exclusively used for making PC and console games. Mobile platforms were first supported in the third iteration of the engine. The latest edition – Unreal Engine 4 – is one of the best tools for making 3D games with rich graphics and smooth, life-like animations. Unreal Engine is based on C++ and has a relatively steep learning curve – but once you get a hang of it, you can make really winning games with the engine.

### 6.2.3. Cocos2d



While this Objective-C-based open source game engine for iOS has lost some of its popularity after the arrival of SpriteKit, the Cocos2d-x cross-platform engine (*in C++*) remains in favour for making casual games. The high-end accelerated graphics support of Cocos2d is one of its several developer-friendly features, while the online community support is excellent too. Apart from C++, Cocos2d has versions available in Ruby, C#, Java and JavaScript. Thanks to the scripting language bindings in Cocos2d-x, app-makers can code in their preferred language and churn out high-quality native apps.

## 6.2.4. Libgdx



Apart from Android, libGDX is also widely used to create custom game apps for iOS/Mac OS X, Windows, Blackberry and Linux platforms. It comes with the general Apache 2.0 open source license – and offers a wide range of support tools for the creation of both 2D and 3D games. The gdxAI artificial intelligence framework is available as an extension with libGDX.

## 6.2.5. MonoGame



This engine is almost ideal for game developers who regularly work with C# and/or .NET. The buzzing community support gives a boost to MonoGame, which can be used for making software for the OS X and Playstation 4 platforms too (*apart from iOS and Android*).

### 6.2.6. GameSalad



With user-friendly visual editors and a simple drag-and-drop mechanism, GameSalad ranks right among the finest game development tools, particularly for all the non-coders out there. Mobile games created with this tool generally have considerable visual appeal. It is compatible with Windows 7-8 and OS X 10.7 and later versions.



## 6.2.7. Marmalade



A C++-based tool suite for professionals from the field of mobile game development. One of the high points of Marmalade is the smooth portability of iOS games to the Android platform – thanks to the Juice tool. Developers coding with the Lua language need to work with Marmalade Quick, while for hybrid app development with CSS/HTML5, Marmalade Web is the go-to tool.

## 6.2.8. Project Anarchy



Released by Havok, the Project Anarchy tool suite has several powerful features, like auto script validation, Scaleform integration and particle lighting support. Developers can also ship their games to Tizen (*in addition to Android and iOS*) with ease – with the help of the free license of Project Anarchy. The Havok Vision Engine powers this tool suite, and the artificial intelligence (AI) and Physics support are both of the highest order. It also supports FBX files in the in-built asset manager.

## 6.2.9. Starling

# Starling

For making the hugely popular Angry Birds mobile game, Rovio used the Starling game development tool. The open-source, cross-platform network has two big advantages – it takes up very little CPU-space and offers high-end code optimization for developers. Sparrow is the iOS-only spin-off of the Starling framework and it lets devs create iPhone apps with Objective-C from scratch.

CocoonJS (for HTML5 games), PlayCanvas, App Game Kit, Stencyl and FMOD Studio are some other high-quality game engines often used by developers.

---

# 7. Mobile Apps For Kids

---

***In this chapter, you will learn:***

- The features that every good kids' app should have.
- 

Present-generation kids are more tech-savvy than ever before. A 2015 in the United States found that, nearly 96% of all kids use mobile devices - with a high percentage of them being regular app-users (games). With rapid advancements in education technology, mobile learning apps have also emerged as an important category apps. As pointed out in the previous chapter, educational apps are the second-biggest category in the Apple App Store - and have a very strong presence in the Google Play Store as well.



## **7.1. The must-have features in an app for children**

Making a successful app for kids places a different type of challenge to mobile developers. Children typically have lower patience levels - and unless an application is interesting enough, attention levels will waver...and the app will soon be abandoned. We will here briefly look at some broad features that any good kids' app must have.

### **7.1.1. Quick to load**

The splash screen of a kids' application should be visible for a maximum of 10 seconds, before the home/first page of the app appears on the screen. No matter how bright and colorful the splash screen is, you cannot expect a toddler to be patiently watching it for a longer time-span. Ideally, you

should include some small animations on the splash screen.

### **7.1.2. Interactive**

There's a world of difference between the types of apps kids and adults love. While a grown-up would find a personal mobile finance manager app or a news reading app interesting, they would seem uniformly boring to a child. Even in a mobile storytelling app for kids, you should focus on including as many interactive features (games, text-highlighting, character tapping, etc.) as possible. Watching a video or reading piles of text on a phone/tablet screen is something no kid enjoys – (s)he invariably wishes to 'be a part' of the app.

### **7.1.3. Background music**

Remember, you are trying to keep your young audience engaged at all times – and audio effects play a vital part for that. Include a soothing, melodious background music (kids should have the option to turn it off, if they wish). There should be appropriate changes in the sounds, depending on the actions of the users. In a reading app for kids, there should be an option to listen to audio-narrations of the in-app stories. A child would

prefer interacting with a virtual companion which ‘speaks’, and not a dumb app!

#### **7.1.4. Easy in-app navigation**

Many developers make the folly of including too many pages/screens in a children’s app. This invariably makes the menu of the app cluttered – and kids ultimately lose their way in the maze. A toddler might be surprisingly tech-savvy, but even then (s)he would appreciate it if the navigation system in a mobile app is smooth and easy to understand. Let’s look at it this way – a kid should not have to run to his/her mom/dad to understand how an app should be operated. Presence of too many screens will make an app heavy too – and that’s another thing you don’t want.

#### **7.1.5. Engaging UI**

Nothing interests a tech-savvy child trying out a new app for kids more than a riot of colors/characters on the mobile screen. The graphic designing themes you implement have to be lively and interesting – and the touch features have to be excellent. The UI should have a nice blend of lifelike displays, and some elements (e.g., the face of a friendly monster) that appear out-of-the-world.

### **7.1.6. Educational benefits**

If you are making an Android/iPhone app for preschoolers, include elements that would add to the overall knowledge pool of the little ones. For instance, in a digital story about interspace travels (and such stories are pretty common in apps), you can include planetary information. Maths puzzles, crossword challenges, and word-making games are also popular in free apps for kids.

### **7.1.7. Reward system**

Any good mobile application for children should have an in-built virtual reward system, so that the li'l darlings can get that sense of fulfillment after their app-activities. If it's a gaming app, you can go for a points accumulation system, reward coins, or other such interesting rewards. For mobile learning apps for kids, there should be token prizes for children who manage to complete letter-writing tasks, maths exercises, and the like. Audio effects – like the sound of applause, or a voice saying 'Well Done!' – can add to a kids' app's charm too.

### **7.1.8. Kid-friendly controls**

The more complicated your app is, the more difficult it would be for a small kid to manage. Make



sure that the kids' app you are working on do not have more than 2-3 different settings. Toggling between the settings/scenarios should be easy. There's every chance that a toddler will incorrectly tap on the screen at any time – the entire app settings should not get altered by that.

### **7.1.9. Parents should like it**

Any responsible parent would like to keep track of what his/her children is doing with smartphones and tablets – and you can facilitate this by including suitable parental control features in the app. For web-enabled apps, there should be a log of sites browsed – which parents can check on a regular basis. Finally, there should be the option to 'lock the app' at certain times.

A mobile game that is too easy won't appeal to a curious, challenge-loving kid, while if the gameplay is too tough – (s)he might simply give up after a few minutes. Maintaining the right balance between this is vital (difficulty levels of kids' games can vary with age groups). Make sure that the little users do not end up making accidental purchases (in apps that have in-app purchasing option).

A mobile app for kids should offer a seamless blend of enjoyment and learning benefits for its young users. Children are starting out on their 'app

journeys' from a very early age - and for them,  
nothing but the most perfect apps would suffice!

---

# 8. Optimizing Apps At The Store

---

***In this chapter, you will learn:***

- The basics of app store optimization or ASO.
- 

Well over 50000 mobile apps are downloaded per minute from the Apple App Store and Google Play Store (combined stat). Apps that get featured at the stores, the ones with uniformly favourable reviews, the applications that have been extensively promoted - they are the ones that show up in organic searches and have maximum chances of registering high download figures.

At the other end of the spectrum, we have the so-called 'zombie apps' (we had briefly talked about them earlier in this ebook). They have almost zero downloads, are not organically searchable, and are almost 'invisible' at the stores. Half-hearted, inefficient optimization strategies are the biggest reason for this fate of these 'failed apps'.

### 8.1.1. Choose the app name with care

Page 123

and it is important to focus on this part of the title. Do a thorough research to find out which keyword(s) your competitors are targeting – and try including them in your app title too. Avoid using any special characters in the name of an app. Make sure that the name you select is not too similar to any existing application (that would only mean more competition).

### **8.1.2. Use high-quality screenshots**

For visual marketing of apps at stores, screenshots play a mighty important role. The App Store allows developers to upload 5 screenshots, while in the Play Store, as many as 8 screenshots can be put up. Irrespective of the platform, select the two most important screens of your apps as the ‘primary screenshots’ (these are the one that would show up in the search results). In the other screens, try to emphasize on the reasons why a person should invest time, money (if it’s a paid app), and space on his/her smartphone, by downloading your app. Highlight the in-app navigation as well.

### **8.1.3. Choose the right category**

Make a mistake here, and your target audience would find it extremely difficult, if not impossible, to

find your app. Go through the list of app categories at the stores, and select the one which fits your application the best. If there is an option of selecting a sub-category, pick the one that would be: a) accurate and b) will not have too much of competition. Stay away from the temptation of wilfully putting your app in a wrong category (*because it has low competition*) – that can lead to penalizations, and your app can be flagged by users.

#### **8.1.4. App store descriptions**

People download mobile apps after skimming through their store descriptions. That, in turn, brings the importance of preparing high-quality, engaging app store descriptions to the fore. In the description, emphasize on how your app would benefit the user and plug a particular requirement/need, instead of simply rattling off the app's features. There is a conception that words like 'free' and 'new' should not be used in app descriptions – but when used in the right context (for instance, 'free iOS multiplayer game'), they can serve as persuasive catchphrases. Android developers should make optimal use of the 4000-character space for writing a proper, detailed description (for iOS apps, the descriptions have to be significantly shorter). Remember, people browsing the app stores on their smart devices can

see only the first 5 lines of app descriptions (before they tap 'More') – and hence, you need to capture the user's' attention from the very start.

### **8.1.5. The role of the app icon**

The icon is, without an exception, the first point of interaction between your app and general users. The icon that you come up with should provide potential users a clear idea of the nature of your application – even before they move on to the app title and description. Ideally, use icons in the .PNG format (1024×1024), and do not put any words, or the app title on it. Many leading app developers prefer adding borders to the app icon, to ensure that the latter looks good against all backgrounds. Use a single prominent theme in the icon, be creative, and be consistent in your usage of the icon on different platforms. For instance, if your app is available for iOS, Android and Windows – users should see the same icon at the different stores. Use the same icon for the app website and social media channels too. It's all about building up a brand for your app. A well-designed app icon can boost downloads by up to 30% – that translates to thousands of downloads!

### **8.1.6. Customized app versions**

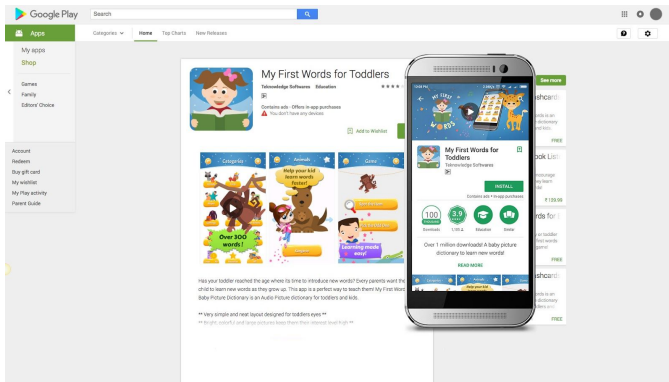
For two identically optimized mobile apps, the one that is compatible with more devices will obviously generate greater downloads. Most high-performing iOS apps have custom versions for the latest iPhones, iPads, iPod Touch and even a Apple Watch extension. On the other hand, Android apps should be tested and made compatible with as many devices as possible. If your app targets a global audience, offer multilingual support on it. That will pull up the download figures too.

### **8.1.7. Use the ‘Keywords’ field (iOS)**

Unlike Google Play, where keywords are tracked from the app title and description only, the Apple App Store has a separate field for entering multiple keywords/tags(max. 100 characters). Do not waste space by adding spaces between the keywords (the section is not visible to general users) or including ‘stop words’ like ‘at’ or ‘the’. While writing numbers, use digits instead of words (i.e., ‘5’ instead of ‘five’). Do not repeat single keywords too many times, which tantamounts to spamming. App store researchers also note that shorter keywords help more in app discovery than elaborate, long-tailed ones.

### **8.1.8. Promo videos (Android)**





Grab the opportunity to enhance user-engagement, by including a short promo video for your app at the Google Play Store. The length of the video should be around 120 to 150 seconds, and you need to include the most important features of the app in the first part of the video. Adding the video is easy (*by linking the YouTube URL*), and it appears in the first position on the app listing page. People browsing Play Store from their mobiles can see the promo video in the top ‘*feature graphics*’ section.

### 8.1.9. Focus on early downloads

The initial volume of downloads go a long way in determining whether your app will be featured at the app stores, or will lag behind as a ‘*zombie*’. At the Apple App Store, the download-count in the first 72 hours (*with maximum weightage on the first 24*

*hours*) after the launch is considered. At the Play Store, you have a little more time – since download data from the first 30 days are taken into account for determining app rankings.

Close to 65% of all iOS app downloads and 58% of all Android app downloads take place through general browsing in the stores. The data clearly shows that, to be successful, an app should be easily ‘discoverable’. And for that, implementation of expert app store optimization techniques are an absolute must.

---

# 9. Success Stories In Businesses

---

***In this chapter, you will learn:***

- Why modern-day businesses need mobile apps.
  - Six interesting case studies on successful implementation of business apps.
- 

Business apps form an important category on both the iOS and Android platforms. As of March 2017, nearly 9.9% of all the applications in the Apple Store were dedicated business applications. By the end of this year, 1 out of every 2 small businesses across the world will have their very own mobile application.

## **9.1. Need for mobile apps in business**

So, how exactly can having a mobile application help in business development? According to survey, corporate houses typically eye three-fold gains from customized apps - a significant increase in sales, a stronger presence in their respective

industry sectors, and an improved experience for clients/users. With competition levels spiralling, having a mobile app is no longer an option for businesses...it is fast becoming a necessity to survive and thrive.

## **9.2. How the implementation of mobile app strategy helped them**

There are plenty of instances of high-profile companies adopting a mobile app strategy - and getting considerable advantages out of it. In the final section of this book, we will take a look at some such interesting case studies for business applications.

### **9.2.1. Pizza Hut Boosts Its Location-Based Marketing**

Nearly 40% of all restaurant apps have a 'mobile ordering' feature. For Pizza Hut - the popular American restaurant chain - orders placed through its mobile application make up around 15% of all orders. A large number of these orders involve home deliveries. With the help of a custom business app, Pizza Hut has been able to deliver greater convenience to its fans worldwide, with prompt responses and order servings. Big-time rival Domino's Pizza has also benefited in a big way

from having a mobile app (which accounts for, on average, 4 out of every 10 orders).

### **9.2.2. Starbucks Enhances User-Engagement and Brand Image**

The official Starbucks mobile app already has around 12 million active users. By early 2016, close to 21% of all transactions with Starbucks were carried out through the application - ensuring greater convenience for general people. The app offers a loyalty program to users, provides real-time information about special offers, discounts (even complimentary drinks), facilitates advance payments (via the 'Mobile Order and Pay' feature), conducts surveys and generates in-app notifications & messages. All of these help in providing a more customized Starbucks experience to the customers.

### **9.2.3. Marble Slab Creamery Witnesses Rise In Customer Loyalty**

Loyalty programs and discount coupons have emerged as must-have elements in mobile business apps. The reason for this is simple enough: it takes approximately 5 times more money to acquire a new customer, than to retain an existing one. Marble Slab Creamery - the Georgia-based icecream chain - serves as a classic

case in point in this regard. With the help of customized loyalty programs, the company's sales figures have jumped by a whopping 60%. Customer spendings have gone up too.

#### **9.2.4. Amazon Offers A Secure Global Mobile Shopping Platform**

The Amazon mobile app can be a reference point for all the other shopping applications out there. People can browse through all the product categories listed, and make purchases. Product wishlists can also be easily shared directly through the app. Buyers can track the status of their orders at any time - and the app also sends helpful push-notifications. It can be safely said that the presence of a smoothly-functioning, uniformly user-friendly mobile application has given a definite thrust to Amazon's business.

#### **9.2.5. Shell Provides A Host Of Important Information To Users**

Leading gas and oil company Shell launched the multi-featured 'Shell Motorist' app in early-2014. Instead of focusing solely on increasing sales, this application brings to users a gamut of accurate, updated information - ranging right from nearest Shell station locators, to the shortest distances to selected destinations. Information about the latest

vehicles, maintenance tips for vehicles and loyalty cards can also be obtained from the app. Shell Motorist efficiently doubles up as a great information app for drivers.

### **9.2.6. easyJet Gives An Edge To Its Flight Services**

The mostly favourable reviews of the easyJet mobile app bear testimony to the quality of service (QoS) it offers. Through the app, passengers can look up flights, make bookings, and even download boarding passes on their devices. For an app that handles a fairly large range of functions, the easyJet app is quick to load, works fast, and ranks high on the reliability front.

Nike, Autoglass, Zappos, UPS and EDF energy are some other examples of businesses that have benefitted in a big way by launching their very own, fully customized, mobile apps.

---











