



香港科技大學

THE HONG KONG
UNIVERSITY OF SCIENCE
AND TECHNOLOGY

MOBILE APPLICATION GOVERNANCE
FRAMEWORK AND GUIDELINES FOR
THE HONG KONG UNIVERSITY OF
SCIENCE AND TECHNOLOGY (HKUST)

Revision History:

Version	Date	Prepared By	Remarks
V0.1	2018-04-06	4D - Jeff Cheung	<ul style="list-style-type: none">• First Draft
V0.2	2018-05-21	4D - Jeff Cheung	<ul style="list-style-type: none">• Second Draft
V0.3	2018-06-22	4D - Jeff Cheung	<ul style="list-style-type: none">• Restructured Content• Enhanced Project Management Content
V0.4	2018-07-23	4D – David Wong	<ul style="list-style-type: none">• Alpha Version
V0.5	2018-07-31	4D – David Wong	<ul style="list-style-type: none">• Beta Version

Content

1. Overview.....	- 5 -
1.1. Purpose of this guideline.....	- 5 -
1.2. Target Audiences.....	- 5 -
1.3. What will be included in this guideline	- 5 -
2. Ownership and Intellectual property right of HKUST Mobile Apps.....	- 6 -
3. HKUST System Architecture	- 7 -
3.1. General System Architecture	- 7 -
3.2. HKUST Web Hosting Solution Overview	- 8 -
3.3. Guidance on selecting appropriate server configuration	- 9 -
3.3.1. Choose Between iHost 2.0 and Virtual Server	- 9 -
3.3.2. Hardware Configuration (For VaaS environment Only).....	- 9 -
3.3.3. Choosing Operating System.....	- 10 -
4. HKUST Mobile App Governance Guideline.....	- 11 -
4.1. Project Member – Roles and Responsibilities.....	- 11 -
4.2. Requirement Guideline	- 13 -
4.2.1. Device Compatibility (iOS)	- 13 -
4.2.1.1. Device Type (Phone / Tablet).....	- 13 -
4.2.1.2. OS Version.....	- 14 -
4.2.1.3. Screen Resolution	- 15 -
4.2.1.4. Supported Orientation	- 16 -
4.2.2. Device Compatibility (Android).....	- 16 -
4.2.2.1. Device Type (Phone / Tablet).....	- 16 -
4.2.2.2. OS Version.....	- 16 -
4.2.2.3. Screen Resolution	- 17 -
4.2.2.4. Supported Orientation	- 17 -
4.2.3. Summary (subjected to change from time to time)	- 17 -
4.3. Functional Guideline	- 18 -
4.3.1. HKUST User Login Requirement	- 18 -
4.3.2. Push Notification Requirement	- 19 -
4.3.3. API Protocol Requirement	- 19 -

4.3.4.	Force update Requirement.....	- 20 -
4.3.5.	App Language Requirement	- 21 -
4.3.6.	App Behavior Tracking Requirement	- 21 -
4.3.7.	App Crash Reporting Requirement	- 21 -
4.4.	General App Design Guideline	- 22 -
4.4.1.	Landing Page / Splash Screen	- 22 -
4.4.2.	Login/Register Screen	- 22 -
4.4.3.	Home Screen / Dashboard.....	- 23 -
4.4.4.	Top Bar.....	- 27 -
4.4.5.	Bottom Navigation Bar	- 28 -
4.4.6.	Side-menu.....	- 28 -
4.4.7.	Setting Page	- 28 -
4.4.8.	General Static Page.....	- 29 -
4.4.9.	App Version Display	- 29 -
4.4.10.	App Icon and App Name	- 29 -
4.5.	Guideline for App Usage and User Engagement Tracking	- 30 -
4.6.	Policy of Collecting Personal Data.....	- 31 -
4.7.	Source Code Provisioning Policy	- 32 -
4.8.	App Versioning Guideline.....	- 32 -
5.	HKUST Mobile App Technical Guideline	- 33 -
5.1.	HKUST CAS Login.....	- 33 -
5.2.	Data Security	- 34 -
5.3.	Push Notifications	- 36 -
6.	API and Integration.....	- 37 -
6.1.	API Protocol.....	- 37 -
6.2.	Firebase Analytics.....	- 38 -
6.3.	Firebase Crashlytics.....	- 39 -
6.4.	Source Code Management and Provisioning	- 40 -
6.5.	App Distribution Guideline.....	- 40 -
6.5.1.	App Distribution Guideline for iOS	- 40 -
6.5.2.	App Distribution Guideline for Android.....	- 41 -
6.5.3.	External Service for App Distribution (Mobile App Management).....	- 42 -

6.6.	App Submission Guideline.....	- 42 -
6.6.1.	Submission Procedure to App Store / Google Play Store	- 42 -
6.6.2.	Bundle Identifier / Package Name	- 42 -
6.6.3.	Distribution Certificate (iOS Only)	- 43 -
6.6.4.	Mobile Provision Profile (iOS Only).....	- 44 -
6.6.5.	Push Certificate (iOS Only).....	- 44 -
6.6.6.	App Signing by keystore (Android Only).....	- 45 -
7.	HKUST Project Management Practice	- 47 -
7.1.	Software Development Life Cycle	- 47 -
7.2.	Project Deliverables	- 48 -
8.	Appendix.....	- 50 -
8.1	HKUST Draft Mobile Compliance List v0.1.....	- 50 -

1. Overview

1.1. Purpose of this guideline

The purpose of this guideline is to provide development guidance, functional checklist, project management best practices and advices for all HKUST mobile app projects.

This document contains all the necessary information to assist parties in developing HKUST mobile apps. In addition, this document can be a reference for HKUST staff to evaluate the quality of their works and make sure that the developers, business owners (HKUST faculties, students or external IT service providers) are delivering an up-to-standard product that compliance with UST infrastructures guidelines, development guidelines and security guidelines.

1.2. Target Audiences

- HKUST academic staffs / stakeholders,
- HKUST students, as well as,
- Related external IT service providers
- HKUST faculties / academic staffs / stakeholders,

For non-technical stakeholders:

They can read through the governance part of this document. Those included all non-technical information which they can understand mobile app related information such as UI/UX design, common requirements, etc.

For technical stakeholders:

They can further read through the technical guideline part of this document. Those included some in-depth knowledge such as general system architecture, some HKUST technical requirements as well as programmer practice in developing the apps.

1.3. What will be included in this guideline

This guideline included the following information:

- Ownership of HKUST mobile app, intellectual property right of mobile apps and server application source code.
- General System Architecture of HKUST mobile apps.
- Governance guideline of HKUST mobile apps.
- Technical guideline of HKUST mobile apps.
- Project Management guideline for HKUST mobile apps.
- Compliance Checklist for reference.

2. Ownership and Intellectual property right of HKUST Mobile Apps

HKUST should make sure that they have the full ownership of every deliverables of HKUST mobile app projects. Those deliverables should include but not limited to the following:

- Mobile app source code
- Server source code
- All related documentations
- Designs and UI

HKUST should have the full authority to:

- Use, reuse and alter those deliverables.
- Transfer those deliverables to another developer to continue the development.

3. HKUST System Architecture

3.1. General System Architecture

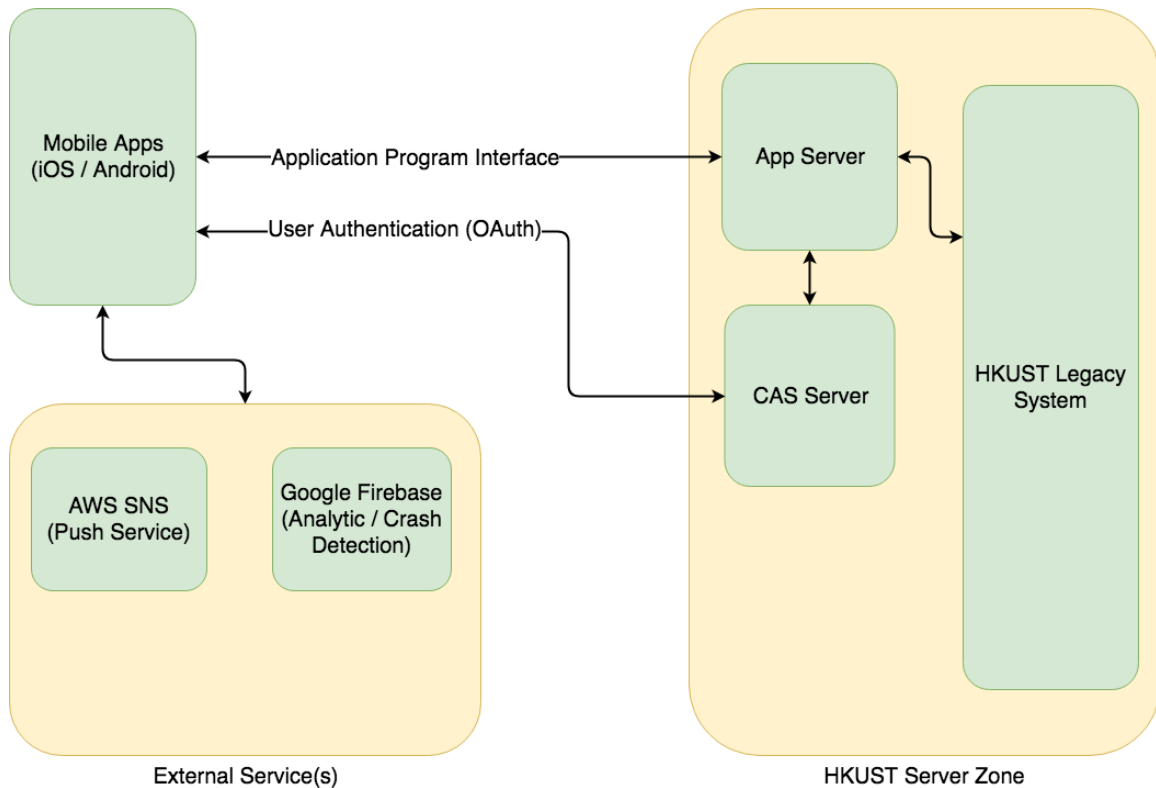


Figure 1. General System Architecture of HKUST Mobile App

Above shows the general system architecture of HKUST mobile apps. The diagram components are described as below:

Name	Descriptions
HKUST Server Zone	This is the server environment within HKUST. Unless exceptional case, all types of servers should be located within HKUST environment, under ITSC supervision.
App Server	The main communication channel with mobile app. Developer should developer APIs on this server. Further to APIs there should be a Content Management System (CMS) which enable users to manage their content through web portal.
CAS Server	HKUST Authentication Server. All HKUST students and staffs should pass through this server before granted access to HKUST related information. CAS Server adopts OAuth 2.0 authentication protocol.
HKUST Legacy System(s)	Those included any existing systems / servers in HKUST campus. Mobile app / app server may need to communicate with those systems / servers in order to acquire any information in need for developing the mobile

	<p>apps.</p> <p>Sometimes the data acquired from those legacy systems need to be stored / cached in app server. In such case, risk of data leakage should be carefully addressed.</p>
External Service(s)	<p>Those are services provided by external parties such as Google, Amazon. Those external services can be any other 3rd party service which can fulfill different objectives depending on the functionalities of the mobile app.</p>

3.2. HKUST Web Hosting Solution Overview

HKUST provides 2 types of hosting website / web application:

	iHost 2.0	Virtual Server (SaaS)
Overview	iHost allows users to build their own websites with database queries in a secure web hosting environment with common website configurations.	Users have full control of rented virtual servers which run on either Windows or Linux2 and are free to install software tools and build programs to fulfil their own needs.
Feature Highlight	<ul style="list-style-type: none"> • Support PHP and MySQL with Apache on Linux • High flexibility to host common website contents and customize website settings • Different from the old iHost, iHost 2.0 allows to run PHP files outside “cgi-bin” folder 	<ul style="list-style-type: none"> • Full flexibility to host any website contents and programs supported in the selected Windows or Linux • Server capacity can be purchased and extended
Best used for	<ul style="list-style-type: none"> • Specific requirements for website contents and features 	<ul style="list-style-type: none"> • Complex websites with specific requirements and custom-made solutions • Dedicated server capacity and performance are required
Technical Expertise Required	Medium to High	High
Maintenance and Responsibilities (ITSC)	<ul style="list-style-type: none"> • ITSC regularly apply security patches to maintain the security level of the hosting environment 	<ul style="list-style-type: none"> • Nil
Maintenance and Responsibilities (User)	<ul style="list-style-type: none"> • Websites need to comply with Cybersecurity Policy and Application Minimum Security Standard • Website owners need to dedicate technical staff (or 	<ul style="list-style-type: none"> • Servers and websites need to comply with Cybersecurity Policy, Application Minimum Security Standard and Server Minimum Security Standard • Website owners need to dedicate

	vendors) as Technical Owners for website maintenance such as resolving security vulnerabilities in the lifespan of the websites	technical staff (or vendors) as Technical Owners for both website and server maintenance such as resolving application vulnerabilities and system patching in the lifespan of the websites.
Data Security	NOT ALLOWED to store HIGH RISK DATA	ALLOWED to store HIGH RISK DATA
Hosting Fee	Free	Refer to HKUST VaaS Hosting

3.3. Guidance on selecting appropriate server configuration

3.3.1. CHOOSE BETWEEN IHOST 2.0 AND VIRTUAL SERVER

Below is the technical reference before making decision on using which type of hosting solution:

- Technical Considerations

If you confirm using PHP / MySQL / Apache for coming project for web applications / website, you can consider using iHost service, which you can left the software installation be handled by ITSC and be more focus on application development. Be sure to check the version of PHP / MySQL / Apache to confirm that they are suit for your requirement. If your web application requires core technologies other than PHP / MySQL / Apache, or if you want to have full control of your application server, you should consider using VaaS environment.

3.3.2. HARDWARE CONFIGURATION (FOR VAAS ENVIRONMENT ONLY)

- Technical Considerations: CPU

Normally, dual-core CPU serves well for majority of web application project. If you are working on complex algorithmic operation which needs high CPU usage, or if you are planning that the server will serve a large amount of concurrent users (over 1000 concurrent users), consider using quad-core CPU instead.

Recommended: Dual-Core CPU

- Technical Considerations: RAM

RAM is the temporary storage for web application. Amount of RAM usage depends on your server OS using (Linux-based OS consumes less RAM while Windows OS consumes more RAM), 4GB RAM usually serves the most purpose of web applications. If you are planning to serve high concurrent connections (>1000 concurrent users), consider upgrading to 8GB RAM or more if you find that the RAM usage is high under live testing.

- Technical Considerations: Storage

The following is accountable for the storage space required for server:

- Permanent storage of database data (if your database is located at the same server as the application server)
- Data contents such as video files / audio files / image files
- Web Application itself
- Backup Image, which stores the server image file.
- Storage reserved for OS paging files

For example, If the web application + database + data contents estimated to use up a maximum of 20GB storage, while we reserve around 40GB for backup image and an extra 8GB for OS paging file. In such case, we expect the minimum required storage would be 20GB + 40GB + 8GB = 68GB and we should consider taking 80GB-100GB as storage space.

- For general purpose, we suggest the recommended virtual server configuration as below:
 - Dual-core CPU
 - 4GB RAM
 - 100GB Storage Space

3.3.3. CHOOSING OPERATING SYSTEM

There are generally two types of OS will be used:

- Windows Server
- Linux-based OS (CLI type, eg. Ubuntu, CentOS)

Normally we choose Linux-based OS as Linux is free-license OS. We will use Windows Server only if specially required by client or the technical solution is based on IIS server (such as ASP.NET, C#).

4. HKUST Mobile App Governance Guideline

4.1. Project Member – Roles and Responsibilities

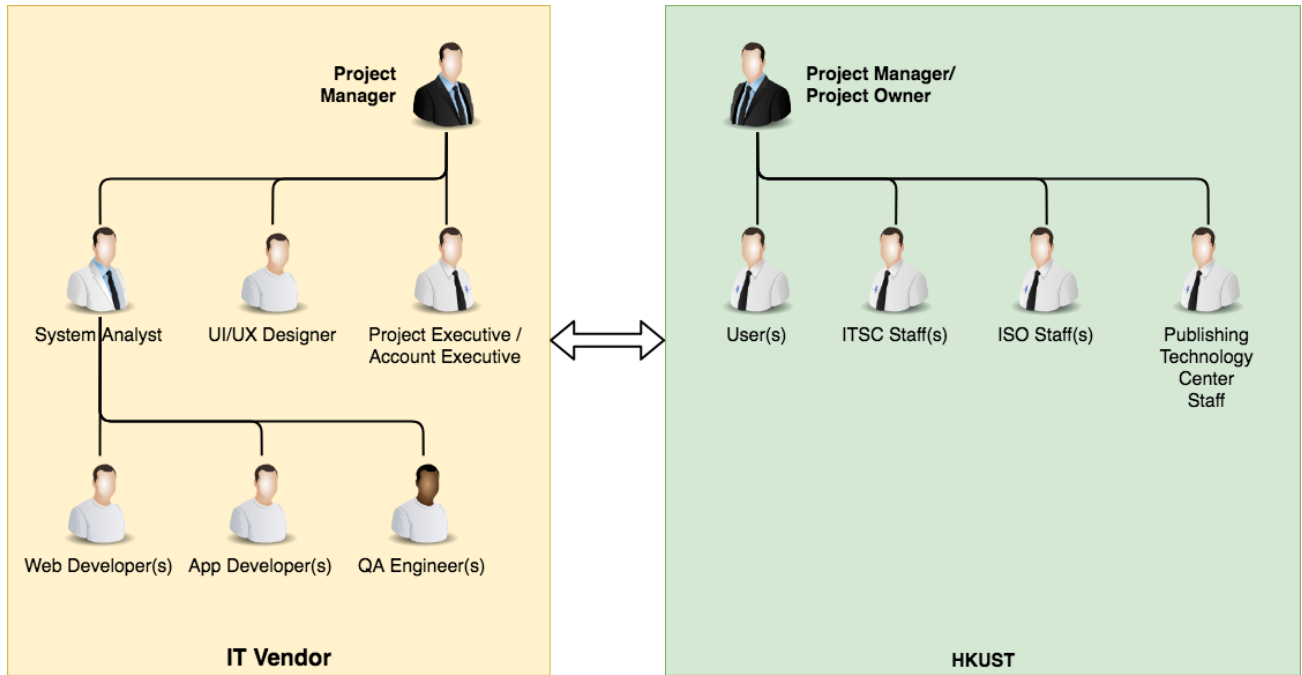


Figure 2. Project Member Structures between HKUST and IT Vendor

Here below explains the roles and responsibilities of each of the member:

IT Vendor	
Type	Roles and Responsibilities
Project Manager (IT Vendor)	<ul style="list-style-type: none"> • Planning of project • Delivery of project and related documentations • Project risk management • Manage project progress • Manage project team members • Manage project changes, evaluate to authorize or reject the changes • Liaison with HKUST project members • Conduction of project evaluation to assess how well the project was managed
Project Executive / Account Executive	<ul style="list-style-type: none"> • Assist project manager in liaison with HKUST project members • Daily project monitoring • Ensure that risks and issues are well-tracked and resolved • Organize project meeting between IT vendor and HKUST
System Analyst	<ul style="list-style-type: none"> • Sometimes this position is replaced by Solution Architect, depending on the scale of the project

	<ul style="list-style-type: none"> • Make decisions on all technical solutions on the project. • Prepare technical documents • Supervise developers for project development • Code-review on developers' work and ensure that the deliverables are up-to-standard
Web / App Developers	<ul style="list-style-type: none"> • Write codes according to technical specifications and functional requirements. • Write test programs to ensure that they meet the requirements of the specifications • Provide appropriate documentation for software programs.
UI / UX Designers	<ul style="list-style-type: none"> • Gathering and evaluating user requirements, in collaboration with product managers and engineers • Illustrating design ideas using storyboards, process flows and sitemaps • Designing graphic user interface elements, like menus, tabs and widgets
QA Engineers	<ul style="list-style-type: none"> • Review technical specifications and user requirements • Prepare detailed and structured test plans and test cases • Executing the test plan to ensure that the project deliverables pass the test cases described • Ensure the quality of project deliverables matches HKUST expectations
Publishing Technology Center staffs	<ul style="list-style-type: none"> • Provide assistance for iOS App launching Management of HKUST iOS Developer Account. (ITSC own Android's account and PTC own iOS)

HKUST	
Type	Roles and Responsibilities
Project Manager / Project Owner (HKUST)	<ul style="list-style-type: none"> • Specify the requirements of project • Engage and manage all other stakeholders (users, senior management, etc) • Sign-off work results • Communicate with IT vendor to facilitate the development progress.
Users	<ul style="list-style-type: none"> • Involve and participate in user requirement discussion • Review and comment on test plans and test cases • Undergo UAT on project • Provide feedbacks and comments on UAT phase.
ITSC staffs	<ul style="list-style-type: none"> • Provide assistance in server establishment for IT vendor • Provide technical advice to assist on project development
ISO staffs	<ul style="list-style-type: none"> • Provide assistance if there is any integration requirement between IT vendor and HKUST legacy systems.
Publishing Technology Center staffs	<ul style="list-style-type: none"> • Provide assistance for iOS / Android App launching. • Management of HKUST iOS / Android Developer Account

4.2. Requirement Guideline

4.2.1. Device Compatibility (iOS)

4.2.1.1. Device Type (Phone / Tablet)

Majority of the app project serves on phone device. If there is no use case that requires tablet device operation, it is suggested to stick on phone device usage only.

If you need the project be tablet-compatible, some consideration need to be carefully addressed such as UI design changes on tablet device.

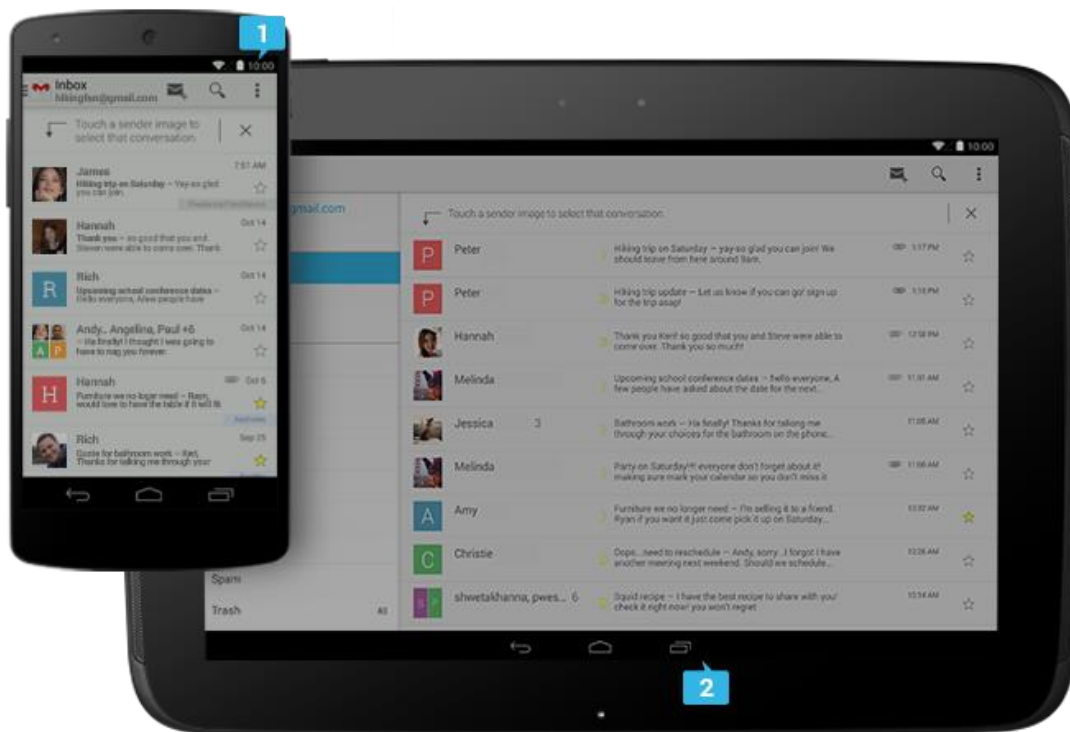


Figure 3. Design difference between phone and tablet device

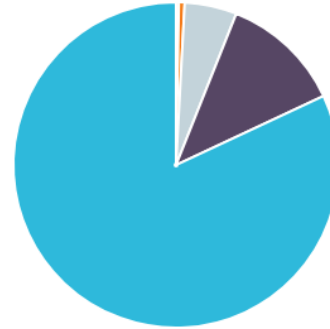
4.2.1.2. OS Version

Below is the statistics of iOS version distribution around the world's active iOS device:

Version	Codename	Distribution
6.0	iOS 6	0.01%
6.1	iOS 6	0.04%
7.0	iOS 7	0.05%
7.1	iOS 7	0.2%
8.0	iOS 8	0.03%
8.1	iOS 8	0.2%
8.2	iOS 8	0.04%
8.3	iOS 8	0.1%
8.4	iOS 8	0.2%
9.0	iOS 9	0.08%
9.1	iOS 9	0.09%
9.2	iOS 9	0.2%
9.3	iOS 9	4.8%
10.0	iOS 10	0.6%
10.1	iOS 10	0.7%
10.2	iOS 10	2.0%
10.3	iOS 10	8.7%
11.0	iOS 11	3.7%
11.1	iOS 11	3.9%
11.2	iOS 11	74.3%

Data Last Updated: 2018-04-05

iOS 6 (0.06%) iOS 7 (0.22%) iOS 8 (0.57%) iOS 9 (5.18%)
iOS 10 (11.99%) iOS 11 (81.91%)



More info can be found on: <https://developer.apple.com/support/app-store/>

For iOS compatibility, we suggest that that app should support up to 2 major versions prior to the most updated major version. For example, if the most updated major version is iOS 11, suppose that the app should support at least from iOS 9, which supports over 95% of iOS user.

4.2.1.3. Screen Resolution

iOS device and their screen resolution are summarized as below:

Device	Native Resolution (Pixels)
iPhone X	1125 x 2436
iPhone 8 Plus	1080 x 1920
iPhone 8	750 x 1334
iPhone 7 Plus	1080 x 1920
iPhone 6s Plus	1080 x 1920
iPhone 6 Plus	1080 x 1920
iPhone 7	750 x 1334
iPhone 6s	750 x 1334
iPhone 6	750 x 1334
iPhone 5, 5C, 5S	640 x 1136
iPhone SE	640 x 1136
iPad Pro 12.9-inch (2nd generation)	2048 x 2732
iPad Pro 10.5-inch	2224 x 1668
iPad Pro (12.9-inch)	2048 x 2732
iPad Pro (9.7-inch)	1536 x 2048
iPad Air 2	1536 x 2048
iPad Mini 4	1536 x 2048

- Some old models such as iPhone 4s or below are not included in the table. We recommend not include those model on the list of device supported as the

market share of those device is very low and this is not cost effective to include those models to the support device list.

- If you are planning to use the app on phone model only, no need to test on iPad device.

4.2.1.4. Supported Orientation

In general, most of the app supports portrait orientation only. Unless special requirement, landscape orientation is not advised be supported.

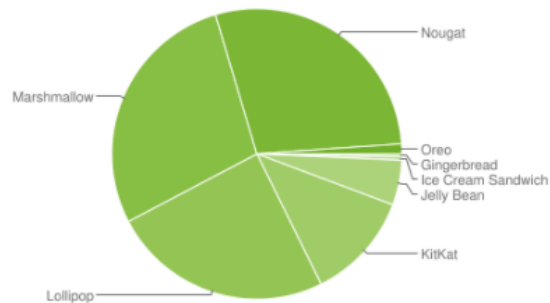
4.2.2. Device Compatibility (Android)

4.2.2.1. Device Type (Phone / Tablet)

Refer to 4.2.1.1.

4.2.2.2. OS Version

Version	Codename	API	Distribution
2.3.3 - 2.3.7	Gingerbread	10	0.3%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	0.4%
4.1.x	Jelly Bean	16	1.7%
4.2.x		17	2.6%
4.3		18	0.7%
4.4	KitKat	19	12.0%
5.0	Lollipop	21	5.4%
5.1		22	19.2%
6.0	Marshmallow	23	28.1%
7.0	Nougat	24	22.3%
7.1		25	6.2%
8.0	Oreo	26	0.8%
8.1		27	0.3%



Data collected during a 7-day period ending on February 5, 2018.
Any versions with less than 0.1% distribution are not shown.

More can be found on:

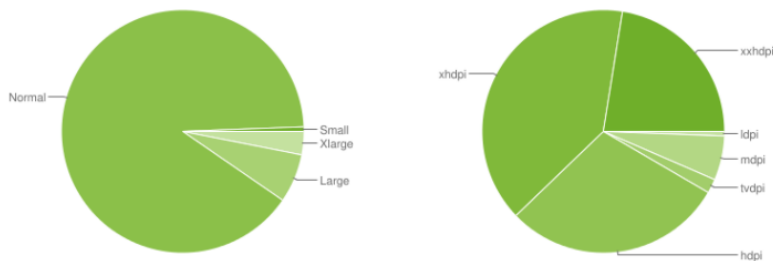
<https://developer.android.com/about/dashboards/index.html>

In general, we suggest the app to support over 80% of the mobile device OS version. In the case of above diagram, we suggest mobile app project should support devices with Android 5.0 or above.

Note that the distribution of OS version varies over time and vendor / HKUST should monitor it from time to time in order to adjust android OS version support policy.

4.2.2.3. Screen Resolution

	ldpi	mdpi	tvdpi	hdpi	xhdpi	xxhdpi	Total
Small	0.5%					0.1%	0.6%
Normal		1.1%	0.3%	28.5%	37.9%	22.0%	89.8%
Large	0.1%	2.8%	1.6%	0.4%	1.2%	0.4%	6.5%
Xlarge		2.0%		0.6%	0.5%		3.1%
Total	0.6%	5.9%	1.9%	29.5%	39.6%	22.5%	



Data collected during a 7-day period ending on February 5, 2018.
Any screen configurations with less than 0.1% distribution are not shown.

Figure 4. Distribution of different screen size on Android device. More info:

<https://developer.android.com/about/dashboards/index.html>

Android device resolution is classified into different dpi type. In general, we suggest all devices with hdpi resolution or above should be supported.

Same as OS version policy, this device resolution policy should be reviewed and adjusted periodically based on the distribution changes.

4.2.2.4. Supported Orientation

In general, most of the app support portrait orientation only. Unless special requirement, landscape orientation is not advised be supported.

4.2.3. Summary (subjected to change from time to time)

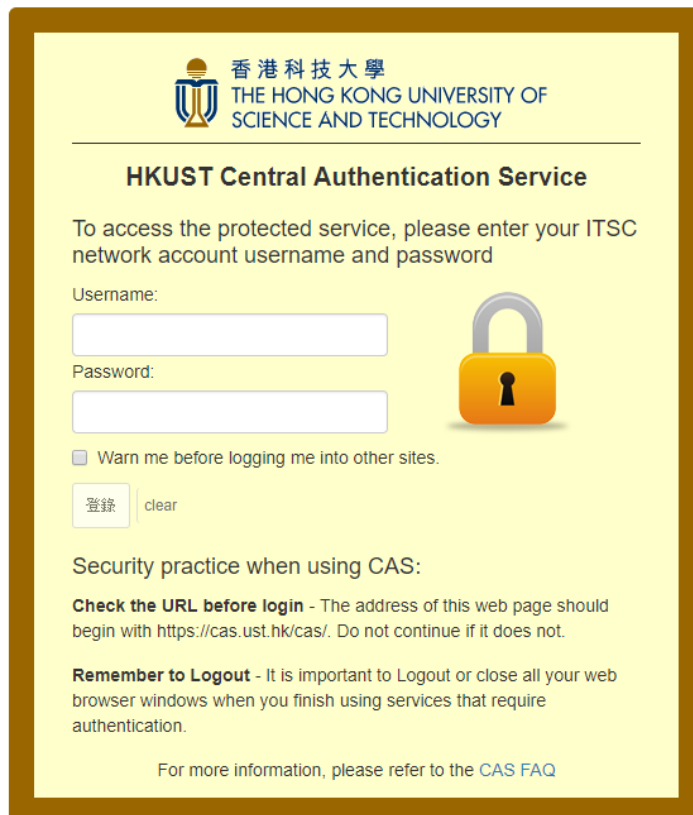
	iOS	Android
Supported device	Phone device only, include tablet if specially required	Phone device only, include tablet if specially required
Supported resolution	Resolutions of iPhone 5 or above	hdpi or above
Supported orientation	Portrait only, include landscape if specially required	Portrait only, include landscape if specially required
Supported OS Version	iOS 9 or above	Android 5.0 or above

Figure 5. Suggested iOS / Android device requirements. (2018)

4.3. Functional Guideline

4.3.1. HKUST USER LOGIN REQUIREMENT

All HKUST member login must use HKUST Central Authentication System (CAS) for user authentication and identification for accessing HKUST related services / information.



The screenshot shows the HKUST Central Authentication Service login interface. At the top, the HKUST logo and name are displayed in both Chinese and English. Below this, the title 'HKUST Central Authentication Service' is centered. A message instructs users to enter their ITSC network account username and password. There are two input fields: 'Username:' and 'Password:'. To the right of the password field is a yellow padlock icon. Below the input fields is a checkbox labeled 'Warn me before logging me into other sites.' and two buttons: '登錄' (Login) and 'clear'. A section titled 'Security practice when using CAS:' contains two paragraphs: 'Check the URL before login' and 'Remember to Logout'. At the bottom, there is a link to the 'CAS FAQ'.

Figure 6. HKUST CAS Login

4.3.2. PUSH NOTIFICATION REQUIREMENT



Figure 7. Example of push notification message

Push notification becomes a popular feature when developing mobile app projects. User can receive updated news and message through iOS/Android device notification panel.

User should provide the following information for push notification requirements:

- Scenario to trigger push notification
- Message body
- Redirect target on tapping the message column (Deep link)
- Target device(s)

4.3.3. API PROTOCOL REQUIREMENT

Mobile app and app server is communicated through a set of API request and response. There are some suggested requirements for API construction and format:

- Restful-API

It is based on representational state transfer (REST) technology, an architectural style and approach to communications often used in web services development.

It is suggested that the development of API should follow RESTful standard which is an open and common standard around the world.

Reference: <https://restfulapi.net/>

- Presentation format: JSON / XML

```
{
  "id": 123,
  "title": "Object Thinking",
  "author": "David West",
  "published": {
    "by": "Microsoft Press",
    "year": 2004
  }
}
```

Figure 8. JSON presentation

```
<?xml version="1.0"?>
<book id="123">
  <title>Object Thinking</title>
  <author>David West</author>
  <published>
    <by>Microsoft Press</by>
    <year>2004</year>
  </published>
</book>
```

Figure 9. XML presentation

JSON and XML are both good data format to be implemented on API response. JSON presentation is well-known to have shorter message length and less overhead. But using XML is not a worse choice if the project picks it as the standard presentation format.

4.3.4. FORCE UPDATE REQUIREMENT

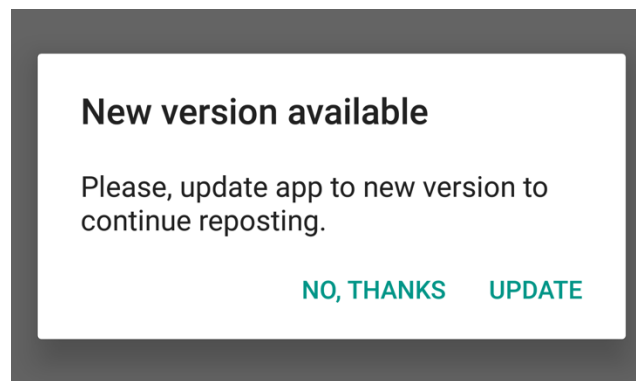


Figure 10. Force update common popup dialog.

App update policy should be implemented so that:

- App user will be well-notified when the app is required to update to new version.

- App user will be forced to update the app when the old version app is no longer usable.

Usually we implementation the force update function classified by the version numbering:

Type of Version Change	Impact	Behaviour
Major version change (eg. 1.2.1 -> 2.0.0)	New version reveals some important functional changes and requires user to update the app to adopt the new features.	User will receive popup at app start to force them to update the app. If app user refuses to update the app, the app will exit immediately to prevent further usage.
Minor version change (eg. 1.1.2 -> 1.2.0)	New version reveals some minor changes on the app where users can have optional choice to update the app to a newer version.	User will receive popup dialog at app start to prompt them to update the app. If app user choose to update the app later on, they can continue using the app.
Patch version change (eg. 1.1.2 -> 1.1.3)	Only some very minor patches are implemented on the new version of the app	User will receive popup dialog at app start to prompt them to update the app. If app user choose to update the app later on, they can continue using the app.

4.3.5. APP LANGUAGE REQUIREMENT

In general the app should provide English language as the preliminary language to app users. It is optional to add language selection to app users to switch between them. Usually English, Traditional Chinese, Simplified Chinese is 3 common app languages to provide for app user selection.

4.3.6. APP BEHAVIOR TRACKING REQUIREMENT

HKUST mobile app should adopt behavior tracking tools in order to well-understanding user behavior in using the app and optimize the UX experience for better user acquisition.

Firebase Analytics is the most popular solution for behavior tracking purpose. For details please refer to section 4.5.

4.3.7. APP CRASH REPORTING REQUIREMENT

To monitor the app stability, HKUST mobile app should adopt crash reporting tools to track every app crash behavior.

Firebase Crashlytics (former known as “Crashlytics” which has been integrated with Google Firebase in 2016), is the most popular crash reporting tools for implement for app monitoring.

4.4. General App Design Guideline

4.4.1. LANDING PAGE / SPLASH SCREEN



Figure 11. A sample landing page

- Landing page / splash screen is strongly suggested to display every time when open the app.
- The landing page design should be tidy and neat. User should be able to identify the app name by a simple glance.
- Suggested landing page display duration will be around 3-5 seconds. Then it turns to login page / main page automatically.

4.4.2. LOGIN/REGISTER SCREEN

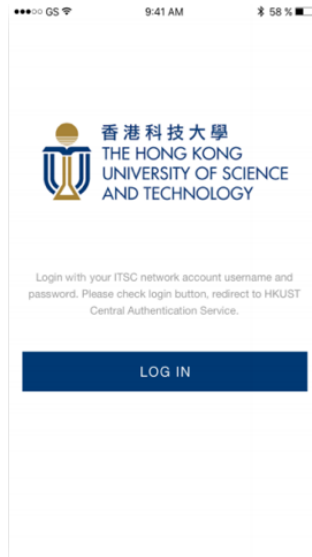


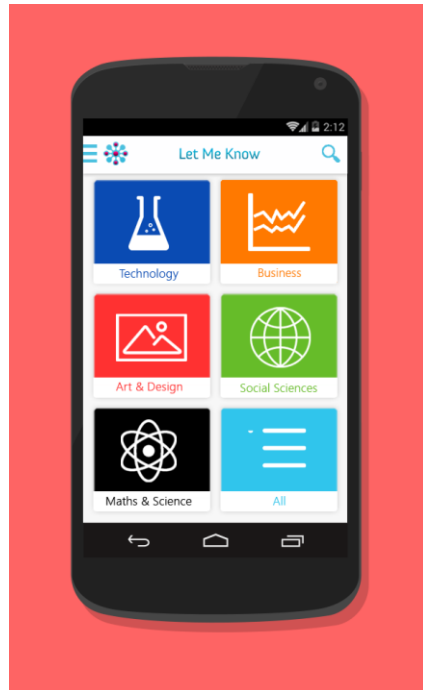
Figure 12. An example of login screen (From UST Research App)

- Unlike other commercial apps, HKUST mobile app should use CAS login as user authentication.
- The login screen should provide UX to redirect the user to CAS login page for user authentication.
- If the user is verified as valid user, the app will be switched to home screen.

4.4.3. HOME SCREEN / DASHBOARD

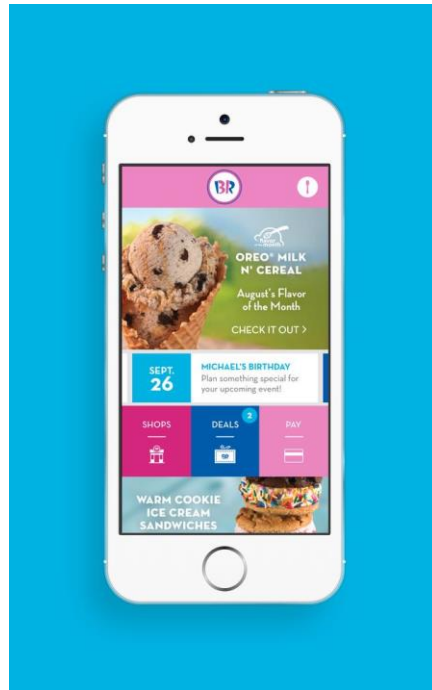
There are numbers of home screen design which can take for reference In designing your mobile app. Here below listed some of the concepts:

4.4.3.1. Grid Item Home Screen



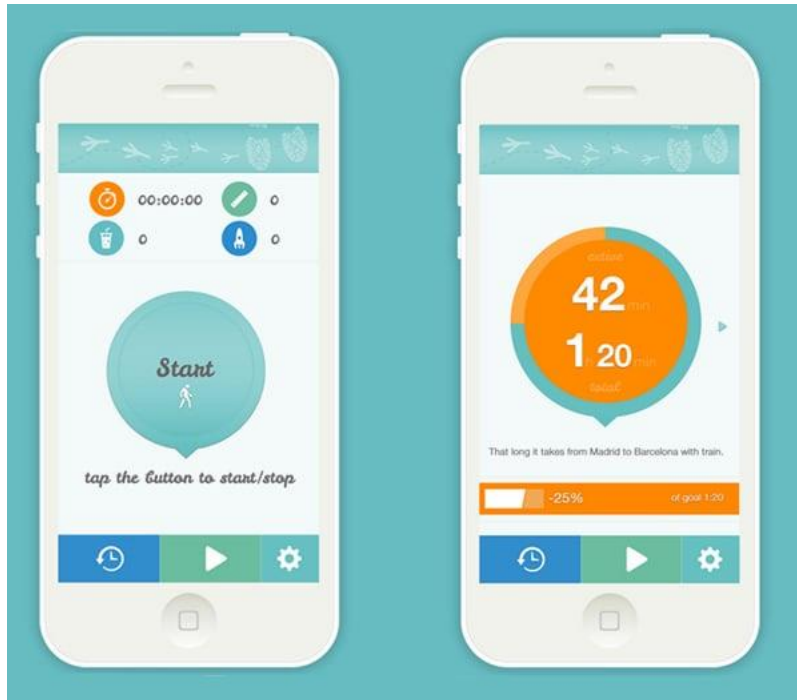
- A Typical home screen design
- Best fit if you have a number of features need to be easily accessible from home screen

4.4.3.2. Banner-rich Home Screen



- Home screen design which is popular among e-commerce apps.
- Choose this design if you need to share information to user instantly, and those data are dynamically change from time to time.

4.4.3.3. Dashboard Home Screen



- Dashboard provide informative numerical data to users.
- Good to use if your app needs to provide instant statistical data to app users.

4.4.4. TOP BAR

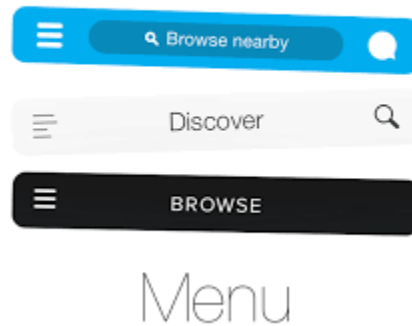


Figure 13. Top Bar

In some app design, a persistent top bar is an essential component to be existed throughout the app flow.

In general, the items in the top bar can be divided into 3 regions:

- Top Center: Display the app name / screen title, or provide a search bar to user.
- Top Left: It should be a side menu hamburger icon, or a back icon if the user enters a inner screen of the mobile app.
- Top Right: There are numbers of icon that can be placed. Notification icon, setting icon, message icon, or search icon. All depending on the app usage.

4.4.5. BOTTOM NAVIGATION BAR



Figure 14. Bottom Navigation Bar

Bottom navigator bar is used for quick shortcut to some feature home page. Bottom navigator bar can be persistent throughout the app journey, or exist only when they are on the home screen of each feature.

4.4.6. SIDE-MENU

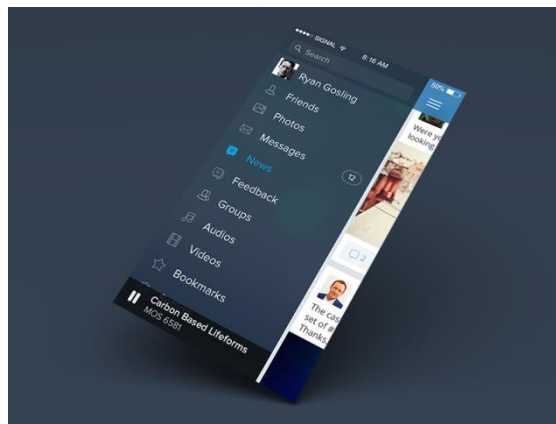


Figure 15. Example of side menu

If you have too many feature items shortcut which should be placed on the app, a better method is to provide a side menu for user to switch between different features.

4.4.7. SETTING PAGE

A setting page is recommended to allow user to change app configurations. Common items in setting page are:

- Language Settings: Options for user to change the UI language

- Notification Settings: Options for user to enable / disable push notification settings.
- Logout: If required, app will provide logout function in the settings page.

4.4.8. GENERAL STATIC PAGE

Some general static page includes:

- About Us / Contact Us: Some introduction about your organization and contact method.
- About the App: Some introduction about the mobile app.
- Terms of Use: Provide information and declaration of HKUST terms of use.
- Privacy Statement: Declaration of privacy policy commitment.
- Related Links / Apps: Provide some related information to HKUST apps or linkage.

4.4.9. APP VERSION DISPLAY

It is strongly recommended that the mobile app should display its mobile app version on the following screen:

- Landing Screen
- Login Screen
- Setting Menu
- Side Menu

The mobile app should display the app version to the above screens, or at least the app version should be displayed on 1-2 screen from the above.

4.4.10. APP ICON AND APP NAME

- *App Icon Design*

App Icon is suggested to be designed using 1536px x 1536px, and resize to corresponding smaller resolution to fit for different device app icon usage.

There is convenient webpage which can auto-generate the necessary icons for usage:

<https://makeappicon.com>

- *App Name Decision*

The general guideline of app name is using around 10-12 English characters in order to fully visible on app drawer.

4.5. Guideline for App Usage and User Engagement Tracking

It is suggested that mobile app should adopt app usage and user engagement tracking tools. A common app tracking utility is Google Analytics for Firebase, which former knowns as Google Analytics.

(PS. Former google analytics will no longer available on 2019. All mobile apps using google analytics should begin to switch to implement firebase analytics for tracking purpose.)



Firestore information: <https://firebase.google.com/docs/>



Figure 16. Google Analytics for Firebase – Dashboard

Guideline for acquiring HKUST account for firebase analytics tracking

Firebase analytics requires user to use Google account to activate the service. For better control of google analytics account:

- The mobile app should adopt firebase analytics through ITSC owned google account, ITSC opens firebase analytics projects and send invitation to department google account / IT vendor's google account.
- If department has its own google account and they wish to manage the firebase analytics themselves, they can also create projects through

department owned google account and send invitation to corresponding IT vendor’s google account.

It is strongly suggested that:

- ITSC should have its own google account for centralized management of firebase projects, and,
- Each department should have their own google account for the same purpose if they have the ability to manage their firebase projects.

Guideline for defining firebase analytics requirement for mobile app

Normally there are 2 types of analytic tracking that mobile app can record:

- Page View
- Event Triggering (Such as tap on a button, or triggering an action to update the database)

We can define the high-level requirement for app tracking first. And then drill in to decide the actual firebase analytics tracking tag.

Here below are suggested formats of high-level requirement listings:

Functions	Tracking Behavior
Page View	Event detail page
Page View	About us page
Event Trigger	Click on the Apply Button
Event Trigger	Click on Cancel Accept T&C Button
Event Trigger	Click on Accept T&C Button

And technically the requirement can be further elaborated as below table

Functions	Tracking Behavior	Event Name	Parameters
Page View	Event detail page	ScreenView	ScreenName: EventDetail
Page View	About us page	ScreenView	ScreenName: AboutUs
Event Trigger	Click on the Apply Button	Apply Event	
Event Trigger	Click on Cancel Accept T&C Button	Cancel T&C	
Event Trigger	Click on Accept T&C Button	Accept T&C	

4.6. Policy of Collecting Personal Data

Office of the Privacy Commissioner for Personal Data has released a guideline on the policy of collecting and using personal data for mobile app usage.

<https://www.pcpd.org.hk/mobileapps/practice.html>

It is required that the mobile app should attach a statement of privacy policy if your mobile app needs user to login before continue using the app, or the app needs to acquire any person data for any purpose. This privacy policy declaration can be a static in-app page, or a redirect link to related webpage.

4.7. Source Code Provisioning Policy

Source Code Ownership and Handover

HKUST should have the full ownership of the source code of HKUST mobile apps. It is IT vendor's responsibility to provide the final version of source code to HKUST before project closing. Also, IT vendor should provide sufficient evidence to ensure that the source code provided can be compiled successfully and functional. If necessary, HKUST can request the IT vendor to provide documentation on code usage.

Source Code Provisioning and Monitoring during development

For better source code monitoring and version control, Git is commonly used as distributed version control system among developers. Examples of popular Git Hosting Services included GitHub and SourceTree.

HKUST technical staff can request to review the code development progress through receive invitation from IT vendor to join the corresponding repository of Git Service.

4.8. App Versioning Guideline

App Versioning policy should be adopted on every IT projects to ensure that the app version is aligned between different parties, especially during the UAT phase.

Semantic Versioning

Semantic versioning is usually used as the format like below:

MAJOR.MINOR.PATCH, increment the:

- MAJOR version when you make incompatible app changes that required the app to be updated (that's where the force update occurs)
- MINOR version when you add functionality in backward compatible manner.
- PATCH version when you make minor bug-fixes

Build Number

Build number increment each time when the app is built successfully. This function is usually based on building script from iOS / Android IDE.

5. HKUST Mobile App Technical Guideline

5.1. HKUST CAS Login

All HKUST mobile app should adopt OAuth 2.0 authentication to verify user's identity before accessing any information. CAS Server will serve as the authentication server to grant user access token for using any of HKUST service.

The general flow of CAS server operation is generalized as follows:

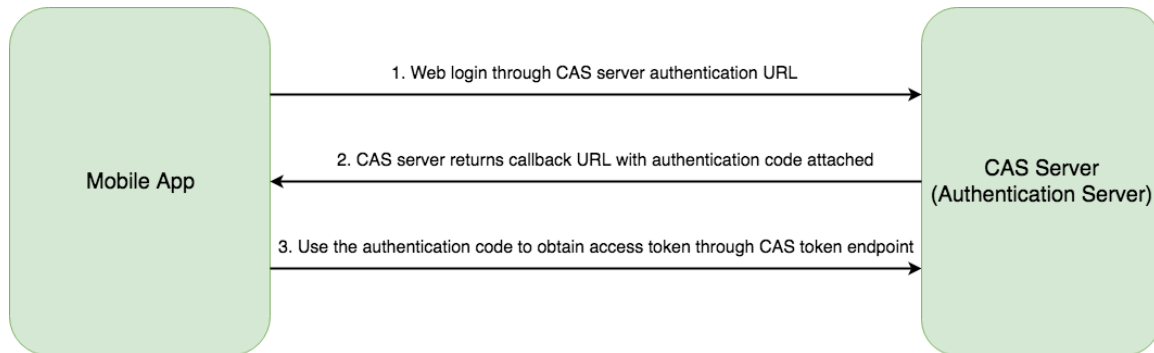


Figure 17. CAS Login: Authentication

1. Mobile app acquires an authentication code from CAS Server through HKUST provided authentication URL.
2. Authentication URL returns with a callback URL with code parameter attached, this authentication code will be used to acquire the access token.
3. Using the authentication code, mobile app obtains the access token through Token Endpoint.
4. You are done! The access token can be used to access any of HKUST resource servers.

Things you should acquire from HKUST (ITSC) for CAS Authentication:

- CAS Server Login URL
- Callback URL
- Token Endpoint URL

The general flow of Application server identification verification is as follows:

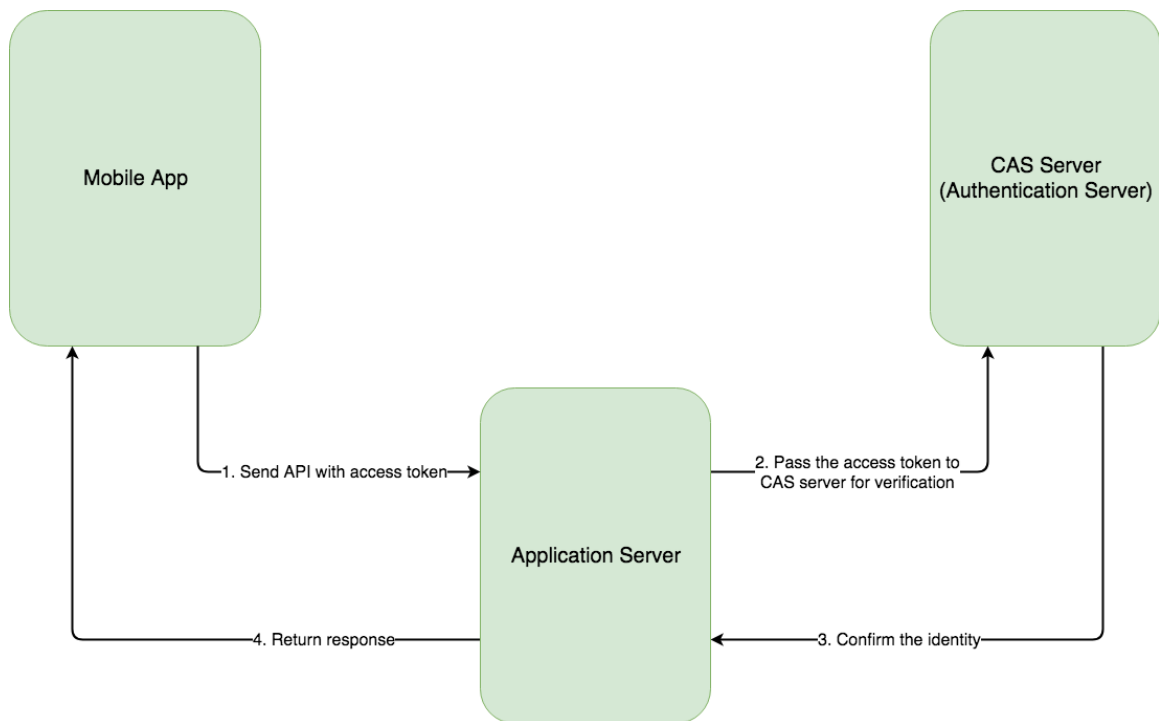


Figure 18. CAS Server OAuth 2.0 Operation: App Server handling

1. Mobile App send API request to app server with access token obtained from previous CAS authentication.
2. App server pass the access token to CAS server and verify its identity through UserInfo Endpoint.
3. App server confirms the identity and looking for resources the API is requesting.
4. App server returns the response with the information that the mobile app is requesting.

Things you should acquire from HKUST for App Server Handling:

- UserInfo Endpoint Information

5.2. Data Security

To ensure data security, HKUST mobile app system should make sure that only minimal necessary information is stored on mobile app and application server. Those sensitive data include but not limit to the followings:

- HKUST students / staff privacy information

For user identity purpose, HKUST email is suggested to be the primary key to identify the user which can be stored in mobile app and application server. For other information, it is

strongly suggested that they should be acquired from CAS server every time when the mobile app / application server needs them.

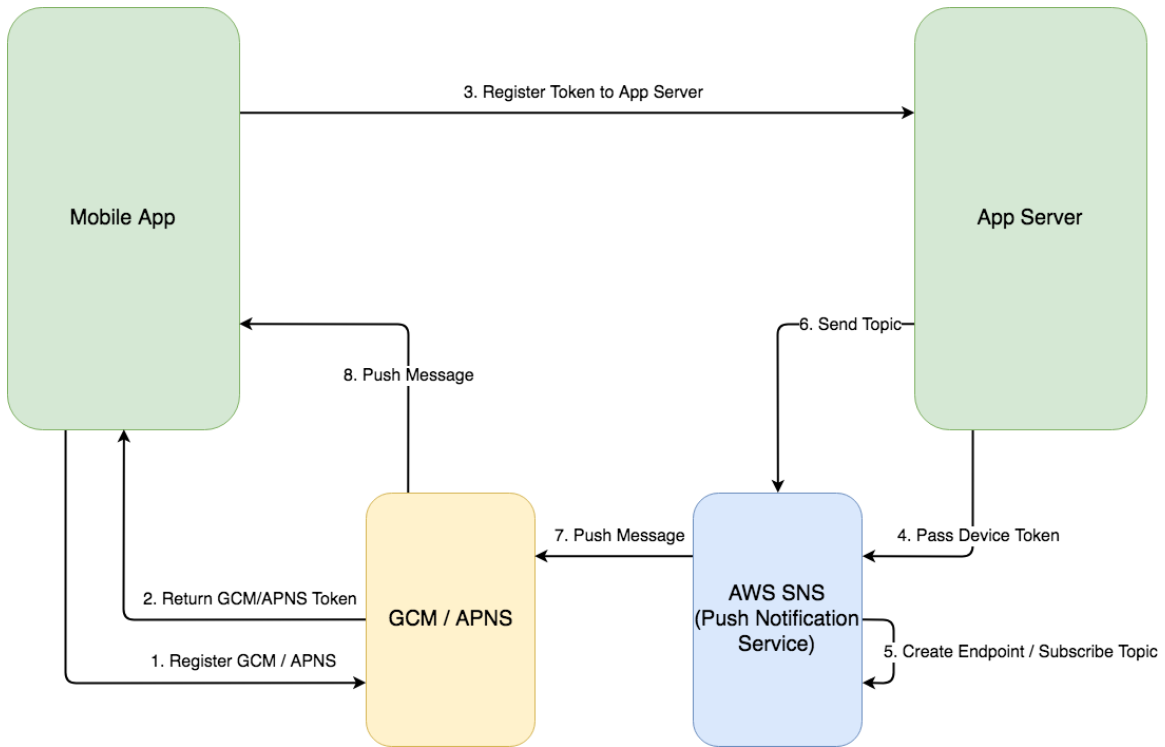
- Data from HKUST legacy server

Those data are usually sensitive information and should keep private and shouldn't be easily be accessed by external environment. Usually we adopt the connection between mobile app and HKUST legacy server in the following way:

- HKUST Legacy server assign access right to app server by adding the app server IP to legacy server white list
- Every time when mobile app needs to access the information from legacy server, the app will send request API to app server.
- App server acquires application token from legacy server and issue API request to legacy server.
- App server will forward the response from legacy server back to mobile app.
- **Please note that during the whole process, app server doesn't store any information returned from legacy server, or app server only stores minimal necessary information from legacy server, to avoid data leakage to external environments.**

5.3. Push Notifications

There are many solutions for push notification service around the world. The document recommends using AWS SNS as push notification service but it is developer's option to choose another push notification service.



6. Figure 19. AWS SNS Flow Diagram

6.1. API Protocol

Mobile app and app server is communicated through a set of API request and response. There are some suggested requirements for API construction and format:

- Restful-API

It is based on representational state transfer (REST) technology, an architectural style and approach to communications often used in web services development.

It is suggested that the development of API should follow RESTful standard which is an open and common standard around the world.

Reference: <https://restfulapi.net/>

- Presentation format: JSON / XML

```
{
  "id": 123,
  "title": "Object Thinking",
  "author": "David West",
  "published": {
    "by": "Microsoft Press",
    "year": 2004
  }
}
```

Figure 20. JSON presentation

```
<?xml version="1.0"?>
<book id="123">
  <title>Object Thinking</title>
  <author>David West</author>
  <published>
    <by>Microsoft Press</by>
    <year>2004</year>
  </published>
</book>
```

Figure 21. XML presentation

JSON and XML are both good data format to be implemented on API response. JSON presentation is well-known to have shorter message length and less overhead. But using XML is not a worse choice if the project picks it as the standard presentation format.

6.2. Firebase Analytics

It is suggested that mobile app should adopt app usage and user engagement tracking tools. A common app tracking utility is Google Analytics for Firebase, which former knowns as Google Analytics.

(PS. Former google analytics will no longer available on 2019. All mobile apps using google analytics should begin to switch to implement firebase analytics for tracking purpose.)



Firestore information: <https://firebase.google.com/docs/>



Figure 22. Google Analytics for Firebase – Dashboard

Guideline for acquiring HKUST account for firebase analytics tracking

Firebase analytics requires user to use Google account to activate the service. For better control of google analytics account:

- The mobile app should adopt firebase analytics through ITSC owned google account, ITSC opens firebase analytics projects and send invitation to department google account / IT vendor's google account.
- If department has its own google account and they wish to manage the firebase analytics themselves, they can also create projects through department owned google account and send invitation to corresponding IT vendor's google account.

It is strongly suggested that:

- ITSC should have its own google account for centralized management of firebase projects, and,
- Each department should have their own google account for the same purpose if they have the ability to manage their firebase projects.

Guideline for defining firebase analytics requirement for mobile app

Normally there are 2 types of analytic tracking that mobile app can record:

- Page View
- Event Triggering (Such as tap on a button, or triggering an action to update the database)

We can define the high-level requirement for app tracking first. And then drill in to decide the actual firebase analytics tracking tag.

Here below are suggested formats of high-level requirement listings:

Functions	Tracking Behavior
Page View	Event detail page
Page View	About us page
Event Trigger	Click on the Apply Button
Event Trigger	Click on Cancel Accept T&C Button
Event Trigger	Click on Accept T&C Button

And technically the requirement can be further elaborated as below table

Functions	Tracking Behavior	Event Name	Parameters
Page View	Event detail page	ScreenView	ScreenName: EventDetail
Page View	About us page	ScreenView	ScreenName: AboutUs
Event Trigger	Click on the Apply Button	Apply Event	
Event Trigger	Click on Cancel Accept T&C Button	Cancel T&C	
Event Trigger	Click on Accept T&C Button	Accept T&C	

6.3. Firebase Crashlytics

To meet with the requirements of app crash detection and tracking, Firebase Crashlytics is a free and advised method of tracking app crash statistics.

<https://firebase.google.com/docs/crashlytics/>

Firebase Crashlytics is a lightweight, real-time crash reporter that helps you track, prioritize, and fix stability issues that erode your app quality. Crashlytics saves you troubleshooting time by intelligently grouping crashes and highlighting the circumstances that leads up to them.

Find out if a particular crash is impacting a lot of users. Get alerts when an issue suddenly increases in severity. Figure out which lines of code are causing crashes.

The installation of Firebase Crashlytics is simple provided that you have experience in applying firebase analytics to your app.

6.4. Source Code Management and Provisioning

For better source code management and progress tracing, app developer should adopt source control policy to all HKUST app projects. Common source control measure we are using is Git technology.

Git is a free and open source distributed version control system designed to handle everything from small to very large projects with speed and efficiency.

Advantages of Git include:

- Branching of different development modules from different developers.
- Tracing of developers' work and their progress.
- Management of source code change history and allow reverting to previous source code version.

Branch Management:

Some general rules of branch management should be followed while using Git branch:

- "master" branch should store the latest source code.
- "develop" branch stores all the developing source code, after completion of development, this branch should be merged to "master" branch.
- When there are multiple modules that are under development, the branch name can use such as "develop-moduleName" for better identification.

6.5. App Distribution Guideline

6.5.1. APP DISTRIBUTION GUIDELINE FOR IOS

For iOS App distribution, the most common method to distribute the app for user testing is through TestFlight invitation.

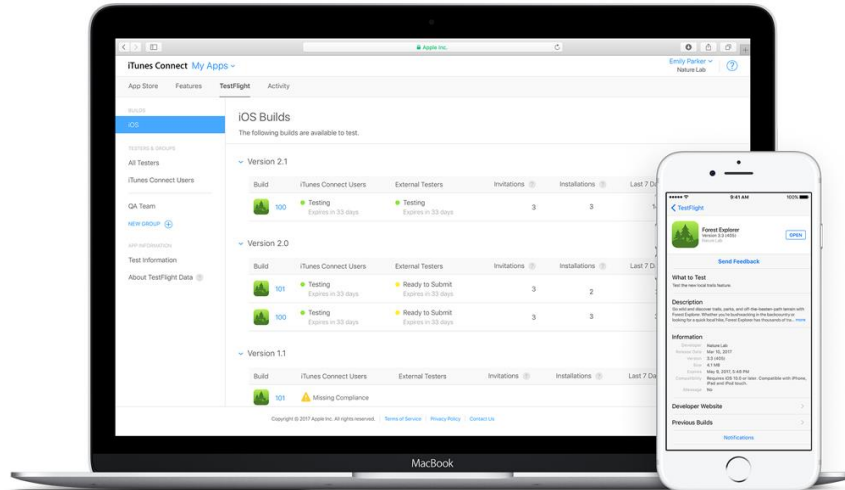


Figure 23. TestFlight Interface

6.5.2. APP DISTRIBUTION GUIDELINE FOR ANDROID

There are 3 methods of app distribution method for testing:

- Provide APK file directly to users for testing. This method is not recommended as it lacks unified management of app distribution version among users.
- Provide a webpage for downloading updated version of app. This is a better method to spread the APK among users.
- Google Play provides Alpha / Beta testing platform for app distribution to users.

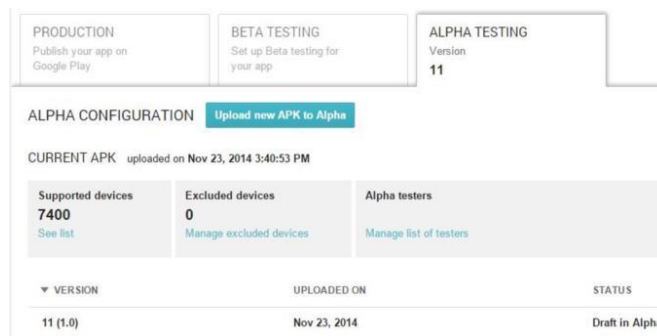


Figure 24. Google Play Alpha / Beta Testing

6.5.3. EXTERNAL SERVICE FOR APP DISTRIBUTION (MOBILE APP MANAGEMENT)

There are still some external paid services (we call them Mobile App Management, or MAM) for app distribution among users. The most benefit using those services is such service can integrate iOS and Android app distribution into one single platform rather than using TestFlight for iOS and Google Play Beta Testing for Android.

Some Examples of External App Distribution Service:

- TestFairy
- Applivery
- HockeyApp

6.6. App Submission Guideline

6.6.1. SUBMISSION PROCEDURE TO APP STORE / GOOGLE PLAY STORE

All the apps owned by HKUST MUST be launched under HKUST App Store / Google Play Account.

It is recommended that the submission procedure to App Store / Google Play Store should be done by technical staffs in HKUST. If submission is required to be done by IT vendors, HKUST should provide the following information to IT vendors for processing:

For iOS submission:

- Bundle Identifier (recommend in the format of hk.ust.XXXXXX)
- iOS Distribution Certificate (.p12 file which is password protected and is exported from keychain)
- Distribution Provision Profile (.mobileprovision file)

For Android submission:

- Keystore file with keystore password, alias and password.
- Package Name (recommend in the format of hk.ust.XXXXXX)

6.6.2. BUNDLE IDENTIFIER / PACKAGE NAME

Bundle Identifier (iOS) / Package Name (Android) is a unique identifier that to distinguish the app. This name is unique and not changeable, unless you need to abandon the original old app and submit it as a brand new app (but you have to

abandon all the user review / download rate / etc, which is not preferable by most project owner).

So, please take extra precautions in creating bundle identifier and package name for your app.

The recommended format of the bundle identifier / package name should be in the form of “hk.ust.XXXXXX”.

6.6.3. DISTRIBUTION CERTIFICATE (IOS ONLY)

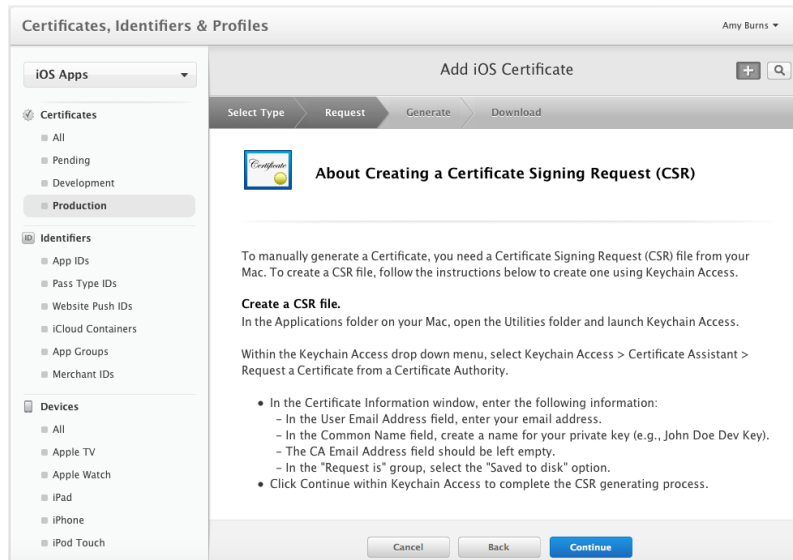


Figure 25. Distribution Certificate Generation

Distribution certificate is required when your app needs to be submitted to App Store for review before releasing for public download. Before that you should upload the private key to Apple Developer in order to generate a new distribution certificate.

To export the distribution certificate for developer usage, goes to Mac Application “Keychain” to export the private key with the certificate together with a .p12 file output.

6.6.4. MOBILE PROVISION PROFILE (IOS ONLY)

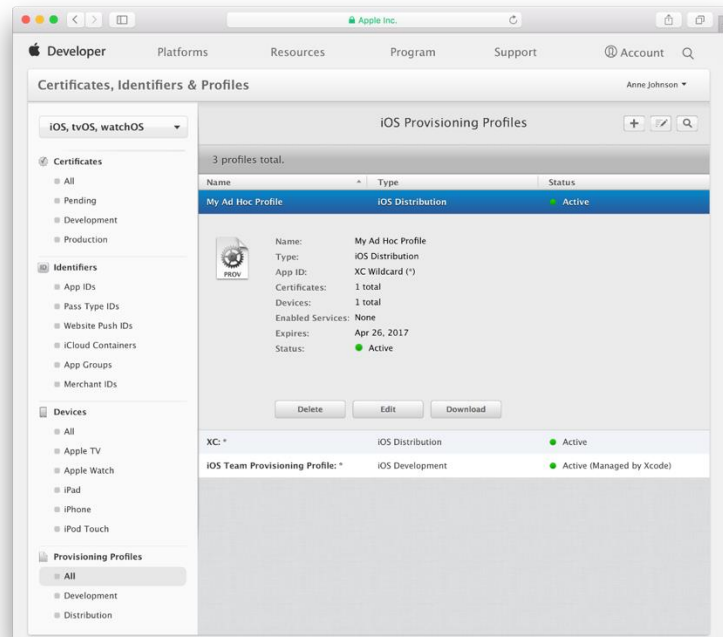


Figure 26. Provision Profile Generation

Distribution mobile provision profile is generated from apple developer -> mobile provision profile page. This provision profile can be directly downloaded from the apple developer page.

6.6.5. PUSH CERTIFICATE (IOS ONLY)

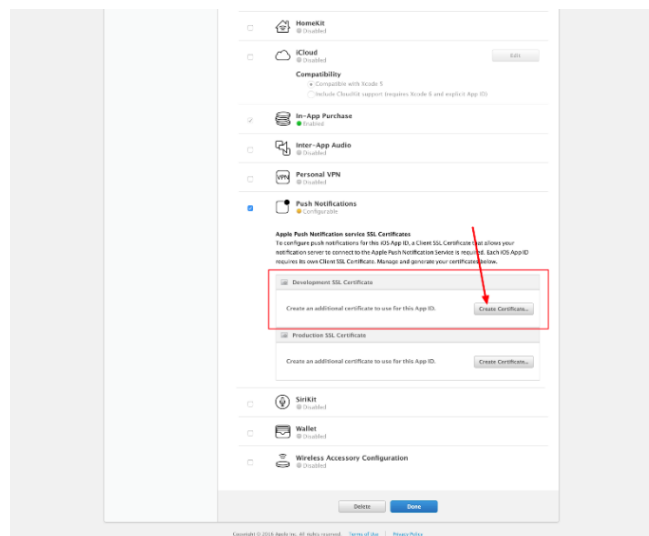


Figure 27. iOS APNS Push Certificate generation

If your app needs to receive push notifications from server, an iOS push certificate must be generated from apple developer account. To create the development push

certificate / production push certificate, simply provide the private key in your Apple developer account and it will generate the push certificate for you.

Note that this certificate should be placed in your server / push service console in order to effectively run the push service.

6.6.6. APP SIGNING BY KEYSTORE (ANDROID ONLY)

Keystore signed Android APK is required for submission to Play Store.

The screenshot shows the 'New Key Store' dialog box. The 'Key store path' is '/home/user/keystores/android.jks'. The 'Password' and 'Confirm' fields are masked with dots. The 'Key' section has an 'Alias' of 'MyAndroidKey', another 'Password' and 'Confirm' pair, and a 'Validity (years)' of 25. The 'Certificate' section includes fields for 'First and Last Name' (FirstName LastName), 'Organizational Unit' (Mobile Development), 'Organization' (MyCompany), 'City or Locality' (MyTown), 'State or Province' (MyState), and 'Country Code (XX)' (US). The 'OK' button is highlighted in blue.

Figure 28. Generate a keystore from Android Studio

It is recommended that every single app should be signed with one individual keystore to prevent leakage of shared keystore for other app's usage.

Business owner should make sure that the keystore file is well maintained and the following information should be recorded for each keystore:

- Keystore filename
- Keystore password
- Alias
- Alias password

If the keystore file or it's relevant information is lost, the associated app will not be able to update to app store anymore. **Please take extra precaution to make sure that the keystore information is well maintained.**

7. HKUST Project Management Practice

7.1. Software Development Life Cycle

The following table illustrates the stages of SDLC.

Stage	Activities
<i>Project Initiation Stage</i>	<ul style="list-style-type: none"> • Establish and confirm project management process and project team formation. • Collate all background information and documentation. • Confirm the business objectives of the project. • Finalize project scope of work. • Finalize project plan. • Confirm project team staffing.
<i>System Analysis & Design Stage</i>	<ul style="list-style-type: none"> • Detail requirement collection and analysis, including consideration of general business requirements, business goals and project objectives. • Architecture design. • Solution design specification production for customer review and comment. • Confirming design specification with customer which will serve as reference within implementation stage.
<i>Implementation Stage</i>	<ul style="list-style-type: none"> • Cloud hosting installation and configuration. • Technical implementation such as software installation and configuration, data conversion, program development, etc. • Executing unit tests and defined QA processes and correct as necessary.
<i>Testing Stage</i>	<ul style="list-style-type: none"> • Validate all solution deliverables. • Conduct an integration test as a rehearsal for go-live. • Carry out User Acceptance Test (UAT) against the validated implementation. • Post-UAT refinements on test result comments and issues.
<i>Go-live Readiness Stage</i>	<ul style="list-style-type: none"> • Project team and business preparation for system go-live. • Defining and agree go-live implementation plan with customer and all relevant parties involved in the project. • Changes in user procedures and documents for training are completed. • Deliver necessary training to the extended user community. • Confirm go-live date. • Dates and schedule for cut-over and implementation are communicated throughout the organization.
<i>System Production Stage</i>	<ul style="list-style-type: none"> • Defining and agree go-live implementation plan with customer and all relevant parties involved in the project. • Deliver necessary training to the extended user community. • Confirm system go-live schedule. • Dates and schedule for cut-over and implementation are communicated throughout the organization. • Execution of defined and agreed go-live implementation plan. • Post implementation check. • First day go-live standby support and system healthiness check.
<i>Documentation and Skill Transfer</i>	<ul style="list-style-type: none"> • Finalize documentation and deliverables to include last minute changes, if any. • Skill transfer on project experience and lesson learnt
<i>Post Implementation</i>	<ul style="list-style-type: none"> • Conduct post implementation review with customer after system go-live. • Post implementation review report production

Stage	Activities
<i>Stage</i>	

7.2. Project Deliverables

Stage/Milestone	Deliverables
Project Kick-off	Project Charter
Project Plan and Proposal Review & Signoff	Project Plan
Requirement Collection and System Design & Analysis Confirmation	Requirement Specification, Detail Functional Specification, Storyboard
Design/Prototyping Review & Signoff	Infrastructure Design Document, Database Design Specification, Application Design Specification
System Implementation and Conduct Unit Test	Database Development Specification, Application Development Specification, Monthly Project Progress Report
System Integration Test and Quality Assurance Test	QA Test Results, Installation and Release Document, Stress Test & Security Assurance Test
User Acceptance Test (UAT)	UAT Plan/ Testing Scenarios Issue Log
Production Rollout and Training	Publish the App to Apple Store and Google Play
Post Implementation	Operations Manual, SLA, Project Summary, Training

Deliverables explained:

Deliverable Name	Description
Project Charter	A statement of the scope, objectives, and participants in a project. It provides a preliminary delineation of roles and responsibilities, outlines the project objectives, identifies the main stakeholders, and defines the authority of the project manager.
Project Plan	Project Plan (or Gantt Chart) describes the purposed schedule of the whole development progress which detailed breakdown from the timing of project kick-off to project closure.

Requirement Specifications	The User Requirements Specification describes the business needs for what users require from the system.
Functional Specifications	Document used to describe in detail for software developers a product's intended capabilities, appearance, and interactions with users.
Storyboard	Document of describing the App Flow with designed screenshots of UI / UX.
Infrastructure Design Document / Database Design Specification / Application Design Specification	Technical document which well described how the requirements / functions can be achieved technically. It usually includes the system architecture diagram, database structure, API documentation, as well as system flow diagram to describe different scenarios.
Project Progress Report	Weekly / Bi-weekly report to describe the development progress.
UAT Plan / Testing Scenarios	UAT Plan describes user acceptance test information which included the UAT test schedule, UAT scope, as well as the acceptance criteria.
Issue Log	This document is for user to fill-in their bugs / issues found during testing.
Operations Manual	App Operation Manual / CMS Manual are the guidelines teaching users how to use the app and the backend CMS.
SLA	Service Level Agreement describes the maintenance scope and the report mechanism during maintenance period.

8. Appendix

8.1 HKUST Draft Mobile Compliance List v0.1

Sections	Compliance Type	Compliance Requirements
1	Basic Criteria	Apps offered in the HKUST Marketplace must meet at least one of the following criteria:
1.1	Run on HKUST Cloud infrastructure	The primary functions or services of the apps must run on HKUST Cloud
1.2	Deployable to HKUST Cloud Infrastructure	Publishers must describe in their offering listing information how the Apps or service is deployed on HKUST cloud infrastructure.
1.3	Integrate with or extend an apps on the HKUST cloud infrastructure	Publishers must indicate in their offering listing information which HKUST cloud apps or integrates with or extends with an existing apps or service integrates or extends on the HKUST cloud infrastructure
1.4	Publishing must be located in HKUST supported by HKUST Marketplace and its partnered or supported organisations and Countries	The HKUST Marketplace current supports the following organisations and Countries..... TBC by HKUST
1.5	Publishers must remain in good standing	The apps or project have to be support soundly by funding or technical sound team
1.6	Publishers offering availability limits	The apps published in the HKUST Marketplace must be of limited or general availability and must be an established target audience base
1.7	Offering in HKUST Marketplace cannot be dependent	The Apps offered cannot use or be dependent on any product or component that is NOT supported or the is no longer commercially available
1.8	Publisher must provide technical documentation	Publisher must be make detailed technical documentation available that describe how to use their offerings on HKUST Marketplace and must provide or link to such documentation in their listing information for each offerings
1.9	Publishing have to listed on HKUST Website	Publisher must announce the availability of their offering in HKUST Marketplace on a public website and must include hyperlinks to their offer listing page(s) on a valid URL.
2	Publishing Offers	
2.1	Timed Publishers Agreement	Publisher must publish at least one offering in HKUST Marketplace within 60 days of executing the HKUST Marketplace publisher Agreement
2.2	Adherence of HKUST Marketplace Technical Requirements	Publisher must adhere to the HKUST Marketplace Mobile Development Guidelines for on boarding as defined in the Marketplace Publishing Guidelines and as may be further identified in the Publishing Portal

3	Offer Listings	
3.1	Publishers Detail Information	Detailed offer information listing pages, which must be accurate and kept up to date. Such information must include, as applicable: - Minimal Offering descriptions; App ID ??, Value Proposition - Recommended Offer Description: Apps ID ??, Value Proposition, Features 3-5 statement, Benefits 3-5 features - Pricing Model: If any - Link to Customer Support information - Offered Resources - User Policy: term of use; privacy policy
3.2	Publisher cannot redirect or upsell	Publishers may not promote or upset to other apps not on HKUST Marketplace
3.3	Publisher cannot promote other Cloud service	Publishers may not promote any Marketplace within HKUST Marketplace
3.4	UST Rights to edit offer page	UST Marketplace reserve the rights to edit or revise offer listing pages details for quality assurance.
3.5	Sensitive Data	HKUST official API that return sensitive data is required to be protected under HKUST Mobile apps Security guidelines. http://itsc.ust.hk/services/cyber-security/mobile-app-security