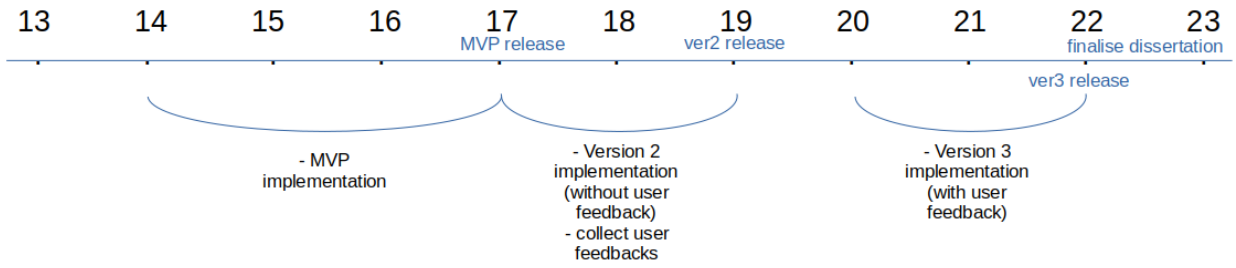
Meeting Agenda

date | 21 – 02 – 2024 (14 : 30)



discussion topics

---

* dissertation ideas
  introduction & background
  - how visualisation helps users to keep track on working out progress
  - how to enhance usability
  - what features make users to keep using the app
  (reading a book about app, I found they are interesting and want to use the knowledge I found when I implement my next version of app, but not sure if they will be relevant)
  - will project execution be like how I implemented all the features regarding the considerations I had?

* cannot use university account currently..
  - couldn’t submit the ethics approval form
  - couldn’t check the emails (outlook)
  (need to check emails and submit the ethics approval form right after I sort out the account)

weekly updates

---

* finalised the MVP
  - implemented basic features
  - data is stored locally
  - able to add / delete customised exercises
  - able to log working out process

* app testing
  - connected testing devices and downloaded the app on them
  - I was testing the app when I go to the gym
  (I have a gym group who are going to the gym together and I am testing my app with them)

* planning next version of the app
  - rather than store data locally, store it on database(connect firebase)
  - enhance usability of the app
  - add visualised muscles

* dissertation
  - wrote about the language I am using to develop the app and why I chose it

priority tasks

---

* muscle visualisation (adobe illustrator – vector path)
* app testing and user feedbacks
* plan development (second version)
* app logo design

Note

* check what is the innovaion case
* git template
* android basic studying : https://developer.android.com , https://developer.android.com/guide
* ads poster
* read study about muscle growth optimisation
* innovation case
* icon

Important dates
poster day week 22
submission week 24
dissertation deadline 9/5/2024 13:00

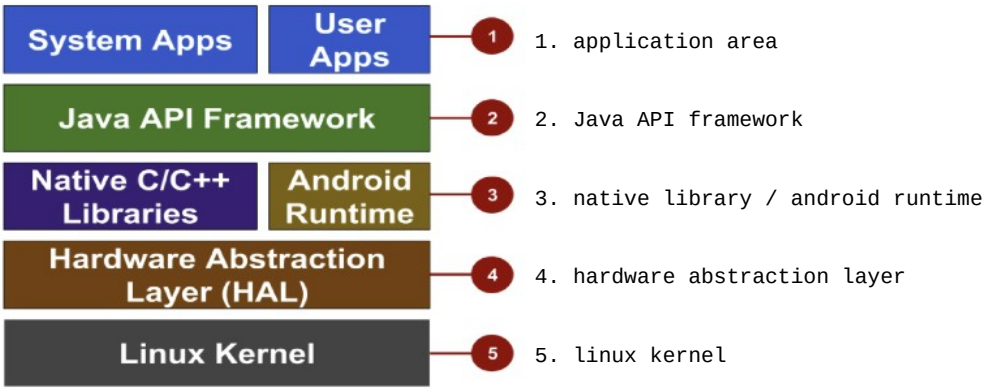
Dissertation ideas
how I developed the app(process), what tools and languages I used, system architecture, scalability, performance optimisation, security concerns, git usage and project management, user friendly interface(why is it user friendly), testing process and user feedback

from meeting :
justify -> why
business
maintaining
launching
value
why/ profit model (why it solves problems well)
framework? why? change?
data type
payment
server
insightful / thesis
show picture as developer!
innovation
clear purpose
reviews
publish and ready to use
what I found interesting
problem solving (what difficulties I met / how I solved it)

MVP
front-end : manually enter data, visualised muscle
(later version, show which muscle to train and stuff)
back-end : store locally, only send data when feedback or report error
(later version, sign up then store it online)

Android
https://dinfree.com/lecture/android/android_1.1.html
* linux kernel
* programmable with java

1.3 android architecture



1. application area
system app : apps which are initially provided
- sometimes invisible
- provide core features of the phone to user app
user app : apps which users can download and are made from developers

2. Java API framework : used when develop user apps
- features that developer can implement are limited by Java API framework
- different android version ↔ different API is provided

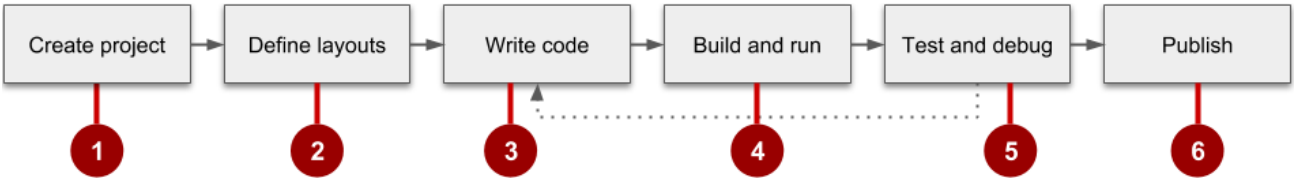
3. native library / android runtime

- c/c++ programs needed to control hardware
(but if you develop an app with c or c++, the program will not be compatible between different devices)
- android is deigned to use java language which uses a virtual machine, which ensures the same program can be run on various different devices
- native library : c/c++ library which enable programs executed via android runtime use hardware
- 자바에서 C/C++ 라이브러리와 연동하기 위한 기술은 JNI(Java Native Interface)라고 하고 안드로이드는 NDK(Native Development Kit)을 통해 구현할 수 있습니다.

4. hardware abstraction layer
- 하드웨어 추상화
- 서로 다른 안드로이드 스마트폰 제조사들의 하드웨어를 동일한 방식으로 접근할 수 있도록 제공되는 호환 인터페이스 레이어
- e.g., 카메라, 블루투스 등 공통 요소에 대한 연결을 보장

5. linux kernel
- 운영체제의 핵심 영역 (커널)
- 스레드, 전원 관리, 보안, 디바이스 드라이버 등 스마트폰을 기본적을 구동하고 사용할 수 있도록 만들어주는 영역

Development process



build tool : compile android project (program source, library, dependency, recourse) → send it to emulator and execute
- maven / gradle(build settings are in .gradle file)

source management : git

API level setting : for app compatibility

Project component

app > java > com.example.helloapp > MainActivity
activity – basic element of android application
one application can be constructed up with many other activities
MainActivity is the start point

app > res > layout > activity_main.xml
define activities' screen structure

app > manifests > AndroidManifest.xml
manifest file defines app's basic properties and register each components of the app
(main setting file for the skeleton of the app)

Gradle Script > build.gradle
script to build the project
define things are needed to build the project such as adding libraries
two files with an identical name
        one is for project (Poject:HelloApp)
        the other is for app module (Module:app)
if the project is made up of many modules, build.gradle will exists in each module

Android component

component – basic element which make up the app
- each component will be the point where users can enter
- a component can be contained in another component
- there are four different types of components (activity, service, content provider, broadcast receiver)

id – for ui widgets in layout xml resource file
name – for resources in value directory

prettify plugin

Muscle recovery