Center for Embedded Computer Systems
University of California, Irvine

< Mobile Embedded Software with Android: Course Design and Experiences >

<Gu-Min Jeong, Dong-Byeong Kang, Sung-Soo Lim and Nikil Dutt>

Center for Embedded Computer Systems
University of California, Irvine
Irvine, CA 92697-2620, USA

gmjeong@uci.edu, gm1004@kookmin.ac.kr
maybeblueday@naver.com
sslim@kookmin.ac.kr
dutt@uci.edu

CECS Technical Report <TR 12-10>
<10> <1>, <2012>

Contents

1. Introduction

1.1. Overview

In this report, we summarize a course 'CS 190: Mobile Embedded Software with Android' (we will use the abbreviated term MESA herein.) which has been offered at UC Irvine in Spring Quarter 2012. CS 190 course is an experimental course and this course has been offered as one of CS 190 lectures.

Generally, in the design of new course, there is a trade-off between both the limitation of university resources and the increasing demand of the new technologies. In other aspect, we should consider a big gap between the industry trend and university course [1]. Also, it should be noted that there are so many students who are already experts in app development.

Considering these, we try to develop a new course which satisfies these kinds of objectives as in Figure 1.1. Especially, we redesign Android programming course (one quarter) to satisfy both the fundamental educational needs and industry trends.

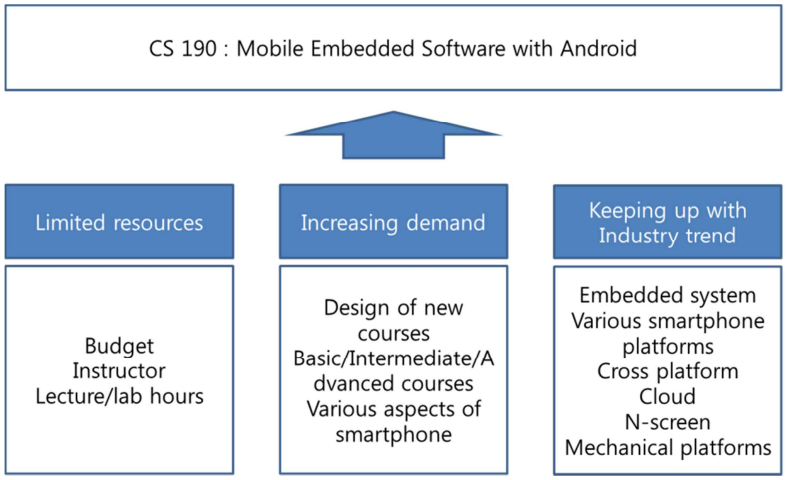


Figure 1.1. Design objectives for the new course design

Consequently, we design an advanced project course in mobile embedded software to meet these multiple goals. Based on the proposed design, we offered a successful course, "CS 190: Mobile Embedded SW with Android" at UC Irvine in Spring Quarter 2012, and describe our experience with the offering.

In this report, we describe our experience in CS 190 and give a discussion for future lectures. We believe that this will be helpful for the course design in many universities and related institutions.

1.2. Summary of CS 190 lecture

Table 1.1, Table 1.2 and Table 1.3 show a brief introduction of MESA provided to the students, weekly plan of the lecture and the people who have been involved in MESA.

Table 1.1 Introduction to the students of MESA

Course	CS 190:Mobile Embedded Software with Android
Number of enrolled students	34 (from freshmen to seniors)
Meeting Information	Room: ICS 180, Day & time: M W F 09:00am to 09:50am
Lab hours	M T W : 4:00-6:00 pm (Not mandatory)
Prerequisite	Knowledge of Java programming, e.g., ICS 21,22,23.
Course Summary.	CS 190: MOBILE EMBEDDED SOFTWARE WITH ANDROID. (4 units). -Mobile embedded hardware and software architectures. -Architecture and implementation of the Android embedded software programming environment for mobile devices. -Application development using Android for smartphones and/or mobile devices. -Project for Android application development for domain-specific tasks such as sensor networks, cloud services, robotics, home networks, etc
Course Overview and Goals	Mobile devices are becoming ubiquitous and are beginning to supplant traditional computing platforms such as laptops and workstations. This course introduces the fundamental concepts of mobile platform architectures (both hardware and software) and uses the Android platform as an exemplar of an open source mobile software platform, upon which a diverse set of applications can be developed. A combination of programming assignments will be used to illustrate and reinforce the concepts taught in class. Each student will participate in a final project that will demonstrate the use of an Android Application on a mobile device for a specific task.
Evaluation	Homework : 40% (8 assignments) Project : 60%

Table 1.2 Weekly plan of MESA

Topical outline	Week 1: Introduction to mobile hardware and software platforms Week 2: Overview of Embedded System and Android Week 3: User Interfaces Week 4: Data management – databases, Files, Content providers, etc. Week 5: Multimedia within Android Week 6: Networking and communications within Android Week 7: Handling Sensors within Android Week 8: Debugging using NDK Week 9: Advanced topics Week 10: Advanced topics

Table 1.3 Instructors, assistants and advisors

Instructors	Gu-Min Jeong - Visiting Associate Professor, Dept. of CS, UC Irvine - Associate Professor, College of EECS, Kookmin University, Seoul, Korea Nikil Dutt - Chancellor's professor and Vice-Chair, Dept. of Computer Science, UC Irvine
Assistants	Dong-Byeong Kang - Graduate student, College of EECS, Kookmin Univ., Korea - Junior specialist, UCI Chang Woo Park - Graduate student, College of EECS, Kookmin Univ., Korea - Junior specialist, UCI Yuhao Phil Ma - Tutor, ICS, UCI
Advisor for GBC(Ginger Bread Car) team	Dr. Nicolas Oros - Research Scholar - Department of Cognitive Sciences
Advisors/helpers for the course materials	Prof. Sung-Soo Lim - Associate Professor, College of EECS, Kookmin University Kyungsoo Lee, Inae Kwak and Truong Huu Phuc - Graduate student, College of EECS, Kookmin Univ., Korea

2. Background and motivation

In this section, we describe the important factors for the course design.
The following aspects should be noted:

- SW platforms and contents become more and more important.
- Platform similarity has increased between industries as well as smartphone platforms.
- Different from the early stage of app market, app market becomes a 'mass market'.

2.1. Trend of recent embedded systems

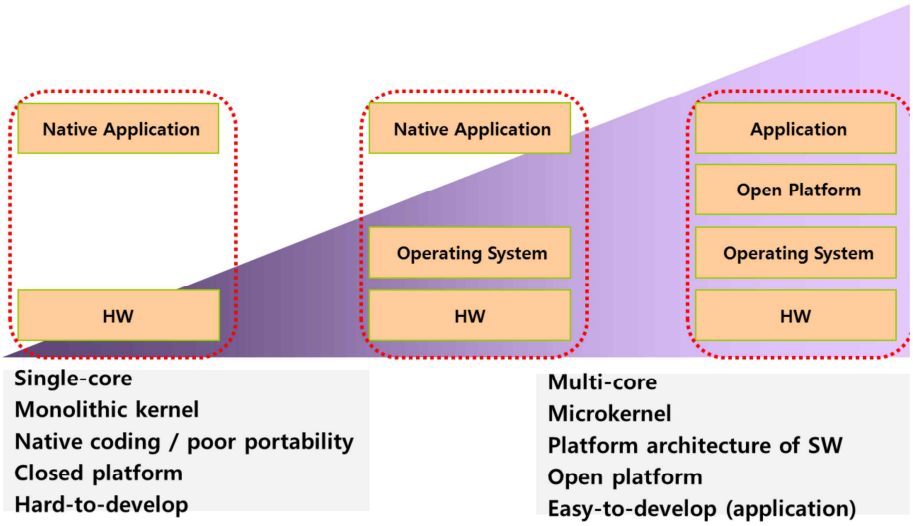


Figure 2.1. Changes in the trend of embedded systems

Figure 2.1 shows the changes in embedded systems. SW platforms and applications become more and more important due to the trend of easy-to-develop tools for applications [2][3][4]. Also, one of the most important things is the platform similarity issue for various embedded systems. The platform similarity for smartphone, PC, TV, automobile, robot and so on has increased rapidly as in Table 2.1.

Table 2.1 Platform similarity between different industries

CPU similarity	Multi-core / safety issues
OS/Platform architecture	Similar platform architecture (layered architecture, etc.) Similar SW modules between platforms
MDA, Simulator and verification	SW architectures become more and more complicated.
Behavioral description	HW modeling for mechanical systems Towards robot and automobile

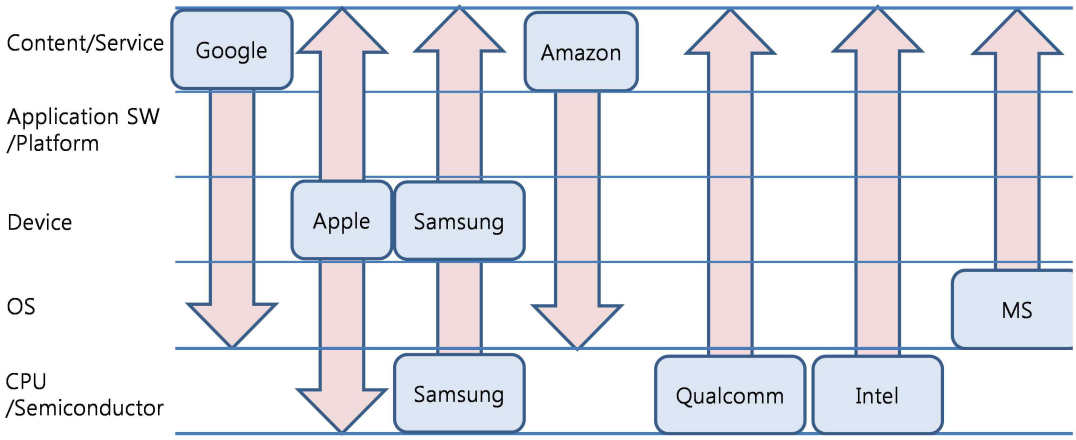


Figure 2.2. Convergence in personal devices

Also, there is a no layer between companies and new value chains are being made by major companies. These markets are changed into a 'mass market' and ''Winner-take-all' market.

2.2. Trend of recent apps

Although a few years have been passed after the successful growth of app markets, there has been a prominent change in the app development. The most important changes can be summarized as follows:

- App market has changed into a 'mass market'.
  - It is hard to get profit for app market for personal developers.
  - 'Winner-take-all' phenomenon happens also in the app market.
- Due to cross platform issues and others, server side development becomes more and more important
  - Smartphone platform becomes just a frame to be used for user interface.
  - Due to the cross platform issues, common module design irrespective of a specific smartphone platform becomes more and more important.
  - General web programming model becomes more and more important.
- Developers should be accustomed to multi-language issues.

Table 2.2 summarizes the changes of app market into a 'mass market'. Accordingly, the competition in app market becomes tougher and tougher. In 'Full Analysis of iPhone Economics', there is a conclusion that 51 years is required for the recovery of initial investment [5].

Table 2.2 Platform similarity between different industries

Changes in App market	Early stage : - Lots of personal applications made by user - Using platform based functions
	Nowadays - Professional apps made by major companies - Using professional engines
Mass market and platform	Major companies just translate their programs into Android instead of re-design. They have their own platform-like things over Android for faster and easier development. It was the same in the previous cell-phone development

Figure 2.3 shows an example of cross-platform issues in messaging apps. Messaging apps can be easily implemented using SMS or Bluetooth based connections and initially there were many messaging apps using this scheme. However, nowadays the server part becomes more important based on the general web based model and hybrid model. Even the SMS messaging service has been replaced by Wi-Fi based messaging apps.

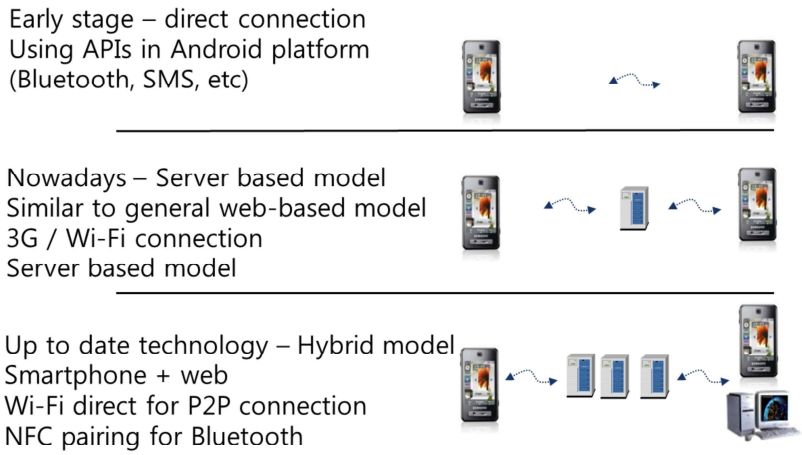


Figure 2.3. Example of changes in messaging related apps

Therefore, we need multiple languages for the development of smartphone apps (as shown in Figure 2.4), and this needs to be factored into development of a new course. For example, for Android programming, we should consider at least Java, C and server programming languages.

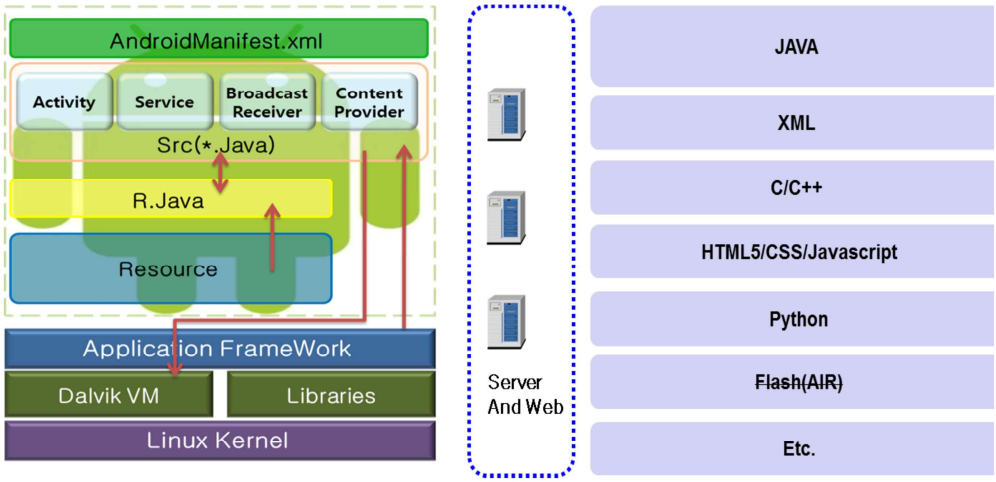


Figure 2.4. Multi language issues in smartphone

2.3. Students' knowledge and basic direction for the course design

As there is an increasing demand for the app related courses, there are many students who are already experts for the app developments. This point is very important for the course design. That is, just offering an introductory lecture for the app development may not be meaningful to the students.

Instead of introducing smartphone platform to the students, we should develop professional examples for the apps and should provide those things to the students. Through these, we can upgrade the app level of the students and save the required time for development. Also, there are so many open source projects and it is very helpful for the lecture.

Providing professional app examples can be considered as providing another virtual platform layer for the app development as in Figure 3.2. Students can use the examples and develop their own apps easily with those examples as in figure 3.2.

3. Course design

3.1. What will be the outcomes of the students

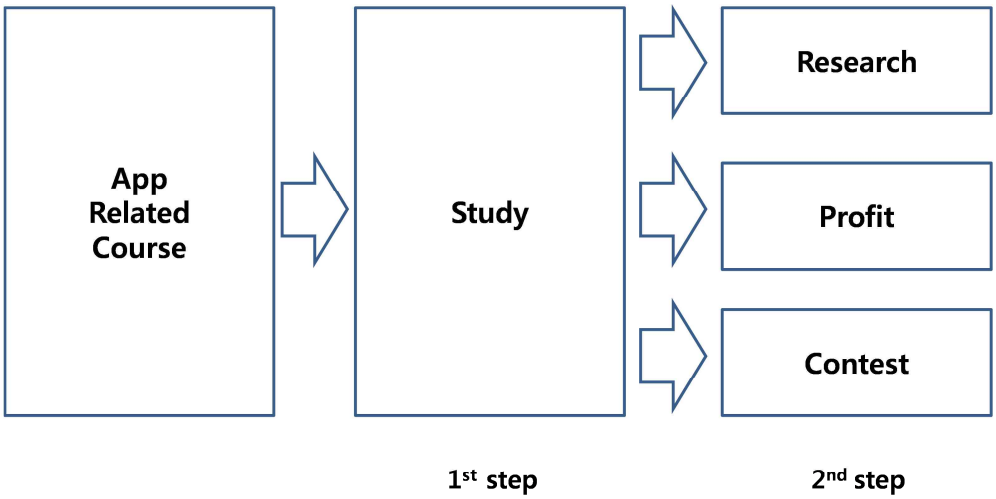


Figure 3.1. Outcomes to the students through app related courses

First, we should consider what kinds of outcomes can be obtained through the app design.

At 1st step, they can study app development itself. They can understand the architecture, APIs, examples for the development of apps. It can be one of the important outcomes of the students.

However, in app related courses, we should consider 2nd step outcomes through the development. That is another reason why we should provide professional examples to the students. We can use the developed apps through the course for paper research, profits in app market and apps the various contests.

Those aspects, research, profit and contest should be considered for the course design.

3.2. Key objectives of the course design

As described earlier, an advanced course on mobile embedded software for smartphones needs to balance the need to inform students of trends in industry, while challenging them with lectures and course materials that foster innovation and creativity.

From the perspective of the smartphone app market and other related industries, the following issues may need to be considered:

- An explanation of user interfaces and platform specific APIs are important
- Highly advanced application designs may be considered.
- Although it is very difficult to cover all the issues of cross-platforms including HTML5, it should be mentioned to the students.

For the development of "killer" applications, many popular open source projects should be analyzed that will help the students during advanced app development.

- It will be helpful to summarize the trend of app market and embedded system related industries.
- From the perspective of a university course, we should consider the following pedagogical issues:
- Design a one quarter (10 week) course, given the limited time and resources.
- The lecture should be helpful both to enhance the knowledge of related courses and to develop highly advanced applications.
- Key mobile embedded system platforms and technologies should be reviewed
- Many sample applications and open source codes should be provided to the students.
- Instructors and teaching assistants should lead the students to develop advanced applications.

Of course, the final design of the course needs to strike a good balance that trades off the needs of industries and the limited resources available in a one-quarter university course. Our main directions for the course design are as follows:

- Supporting enough application examples which can be used directly for the students' application design,
- Analyzing popular open source projects for Android,
- Providing enough knowledge for other related courses such as operating system, embedded system, network, communication, multimedia, data base, etc.,
- Introducing recent trends for mobile devices, including cloud computing, optimization and energy issues, cross platform issues including conversion tools, HTML5, Adobe PhoneGap and Mozilla Boot-To-Gecko,
- Summarizing recent trends of related industries such as smart cars, smart robots and others.
- Preparing sufficient practice examples and homework materials to ensure that students have hands-on, and detailed experience in various aspects of mobile embedded software development for smartphones.

As mentioned in [6], example source code is very important for these kinds of lectures. If source codes of various professional apps are provided, we can upgrade the app level of the students and save the required time for development. Conceptually, we can think that there is another virtual platform layer for the app development as in Figure 4.

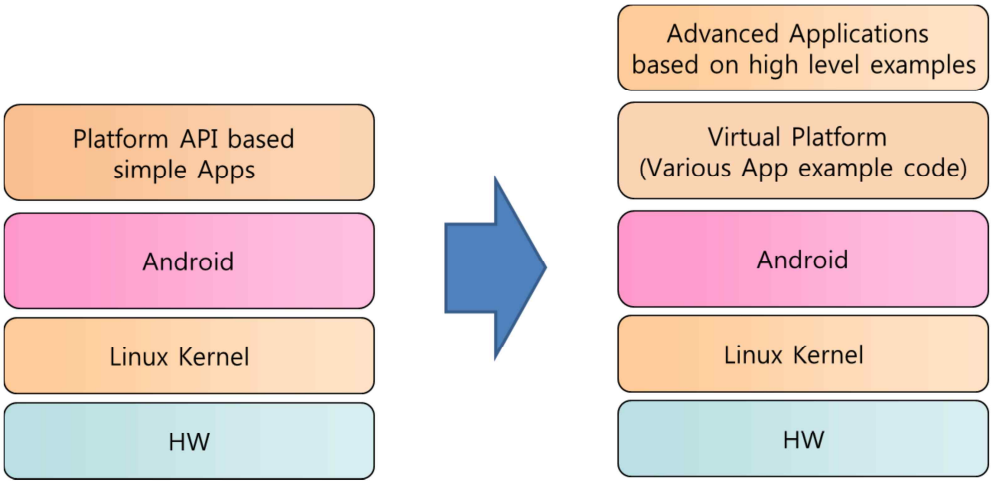


Figure 3.2. Enhancing the level of apps with virtual layer based on the source code of professional apps

3.3. MESA design

The CS 190 course "Mobile Embedded Software with Android" offered at UC Irvine consists of 3 hours of regular lectures (e.g., 1 hour every Monday, Wednesday and Friday) and lab hours for homework. The overall structure for the 10-week course is shown in Figure 3.3.

On Mondays and Wednesdays, we focused on the theory, trends and basic programming skills. On Fridays, we discussed app examples developed by lab members and open source projects related to Android programming through Hands-On-Tutorials (HOT).

In the first 4 weeks, we covered the basics of Android programming. During Week 1 and Week 2, introduction to the course, embedded systems, Android, app trend and Android overview was given. For the HOT and homework, AndSync app, which deals with a synchronization of two threads, was used in Week 1, and BMPCompare app, which compares the execution time of an image rotation engine when using Java program, Android API and C-based native programming with NDK(Native Development Kit), was given in Week 2.

During Week 3 and Week 4, basic UI programming and other essential items for Android programming were presented. For HOT and homework, we gave layout design of a drawing app and a block game, animation using Surface view and other view examples in Week 3, and address book, block game and basic drawing app in Week 4, respectively.

From Week 5 to Week 7, more advanced issues for the development of apps were discussed, including multimedia, network, LBS and AR. On Monday, related trends and basic theories were presented for multimedia, network/communication, LBS, AR, etc. On Wednesday, Android APIs for network, multimedia, LBS, etc. and related app developments were covered. On Friday, we analyzed and explained various developed apps (e.g., Wi-Fi based Chatting, Bluetooth/SMS based message transfer, Basic LBS app, Sensor app, etc.) and open source codes used for app development (e.g., Ringdroid, FFMPEG, Cococ2D, AndAR, NyAR, etc.)

In Week 8, we introduced important issues for the advanced app development, which are NDK, debugging and performance analysis. As exercises, we covered various NDK based applications and debugging/performance analysis tools.

Up-to-date topics were introduced in Week 9 and Week 10. We aim to explain challenging keywords and recent research directions such as cloud computing, N-screen, energy/power issues, performance analysis, optimization, robotics platforms (ROS, OPRoS, etc.), cross platform issues and HTML5 related issues (HTML5, PhoneGap, B2G, WAC, etc.) .

A major component of the course was a final design project structured in groups of 2-3 students per project team. Each project had to propose ideas early in the quarter, and maintain weekly progress updates with periodic reports that were graded. During the finals week (Week 11), we organized a major showcase titled "AMASE 2012 @ UCI: A Mobile Application Showcase Event [7], where with significant industry sponsorship the students display their projects. The best projects were awarded prizes (cash and gifts) donated by the sponsoring companies.

Figure 3.3. A course design for advanced Android programming

3.4. Week 1

3.4.1. Overview

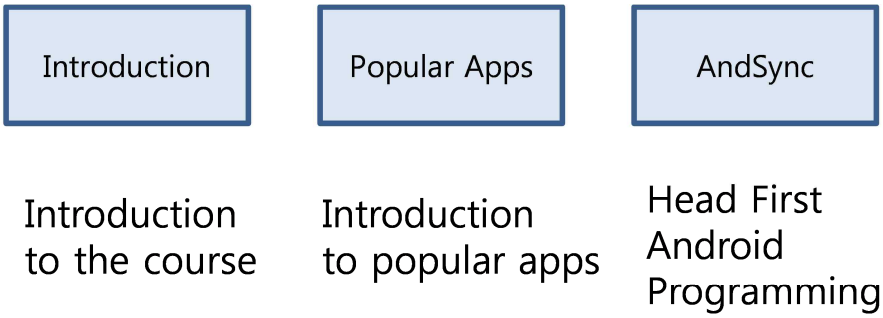


Figure 3.4. Overview of week 1

In Week 1, introduction to the course, recent App trends and an App example for the beginners were given as in Figure 3.4.

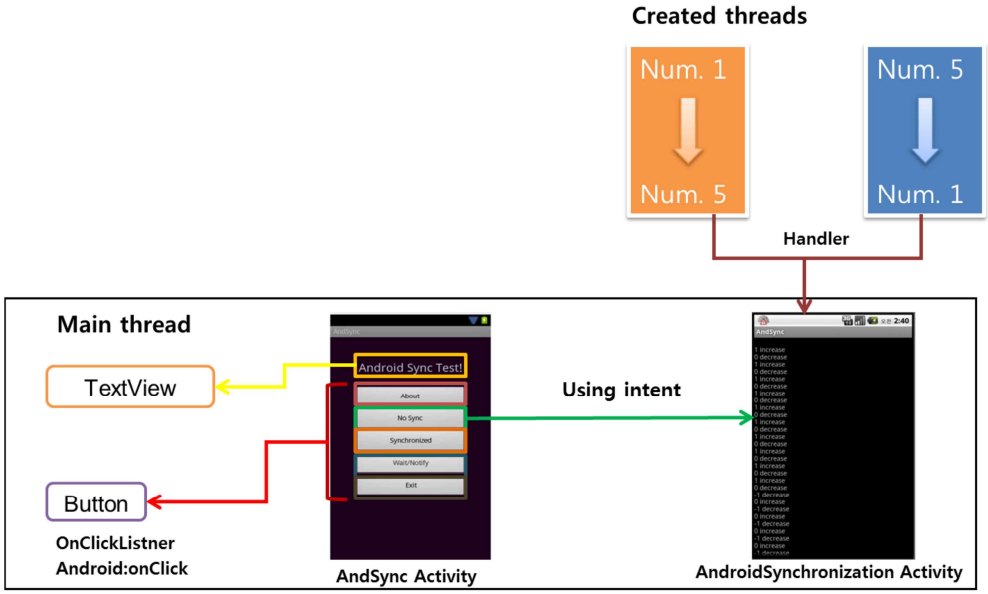
Table 3.1 Overview of week 1

Monday	Introduction – Instructors and courses Examples – Project examples
Wednesday	Apps - Popular Apps Research using Apps - Platform, Multimedia, Network, Sensor, AR, LBS, OPRoS, ROS, etc.
Friday	H.O.T. : AndSync app
Lab hours	Tool installation
Homework	AndSync app

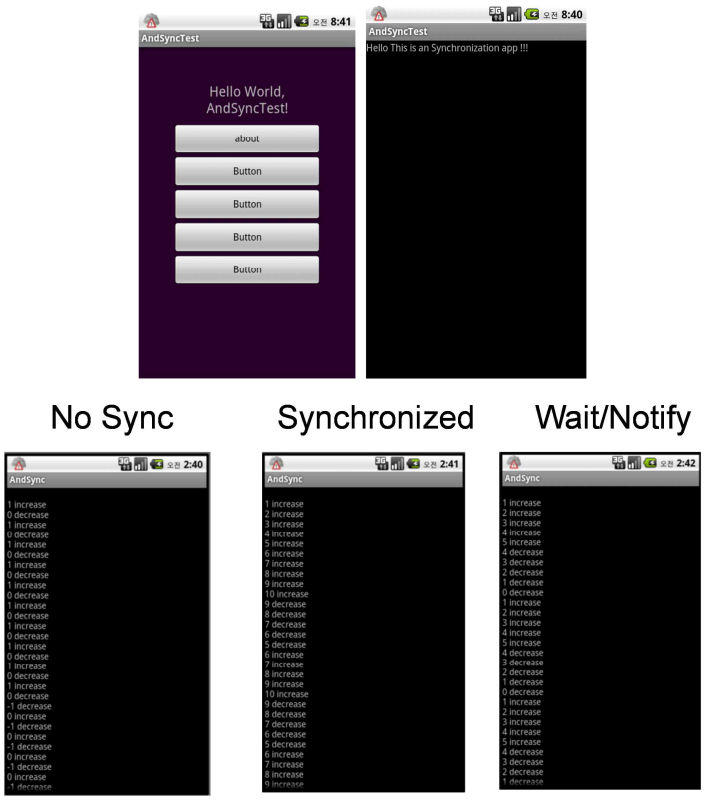
3.4.2. Exercise and homework

As an exercise on Friday, HOT with AndSync App (Figure 3.5) was given. This is an introductory App that helps students understand how an Android App works. Java synchronization program between two threads has been converted into an Android app.

Since this App deals with the synchronization of two threads, it can be connected to the embedded system or OS issues. The app was structured so that Students can follow the instructor to finish the App during the HOT session.



(a) Structure of AndSync app



(b) Screen shots of AndSync app

Figure 3.5. AndSync App

3.5. Week 2

3.5.1. Overview

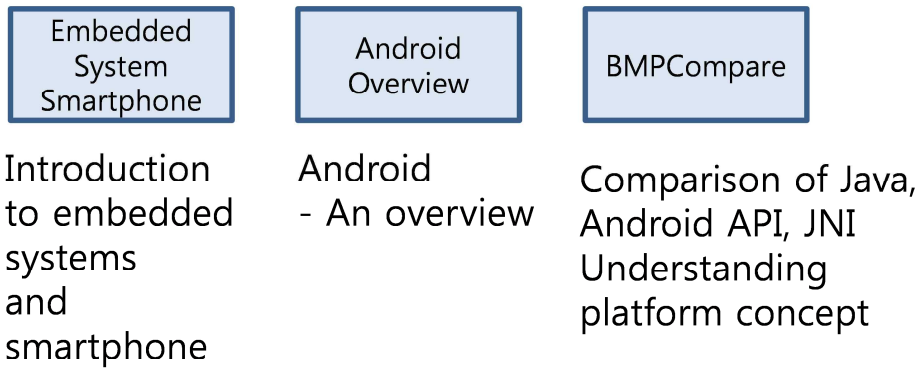


Figure 3.6. Overview of week 2

In Week 2, we introduced the basic concept of the embedded systems, recent trend of smartphones, automobiles and robots, and Android platform overview [8].

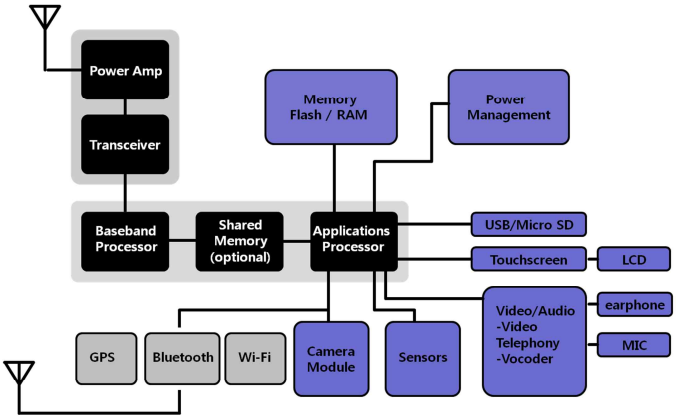
Table 3.2 Overview of week 2

Monday	Introduction – recent trend of embedded systems and smartphone
Wednesday	Android programming overview
Friday	H.O.T. : BMP Compare
Lab hours	Homework of Week 1
Homework	BMOCompare

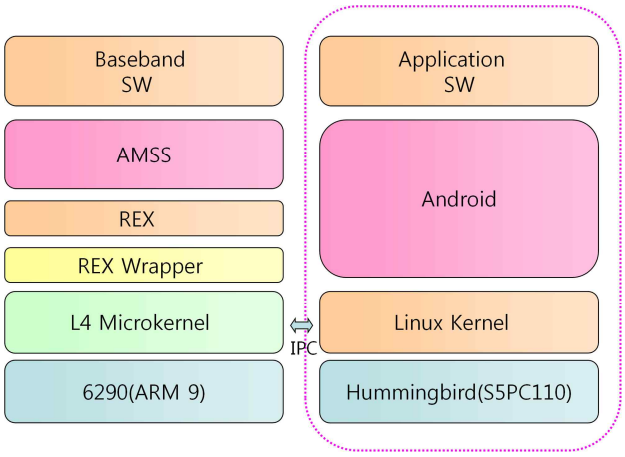
3.5.2. Exercise and homework

Figure 3.7(c) shows BMPCompare app, which is an image rotation example. This is a simple computer graphics example that explains the platform aspects of Android API by comparing the execution time of a Java program, an Android API and a Native program.

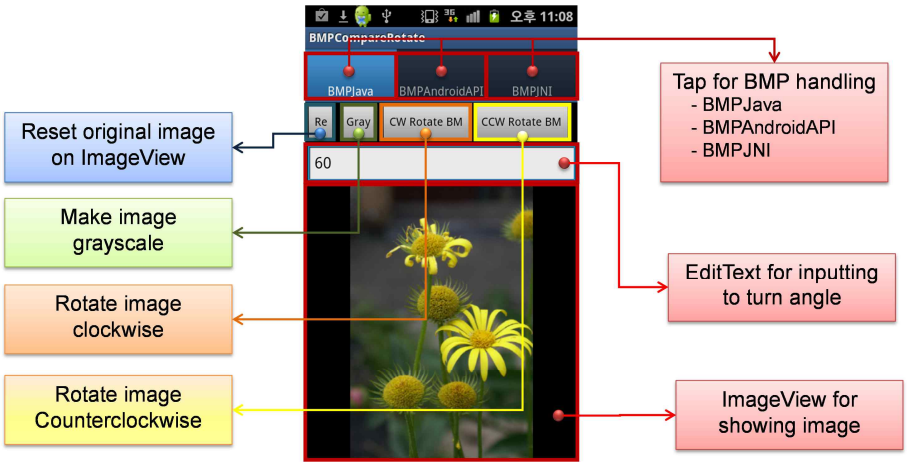
If there is no suitable engine for Native execution, we should implement a new method using Java or C code. Due to the resulting impact on the execution time of Android Apps, students can understand the needs of NDK in Android programming.



(a) HW structure of the smartphone



(b) SW structure of the smartphone



(c) Image rotation example

Figure 3.7. Smartphone HW/SW structures and an image rotation example

3.6. Week 3

3.6.1. Overview

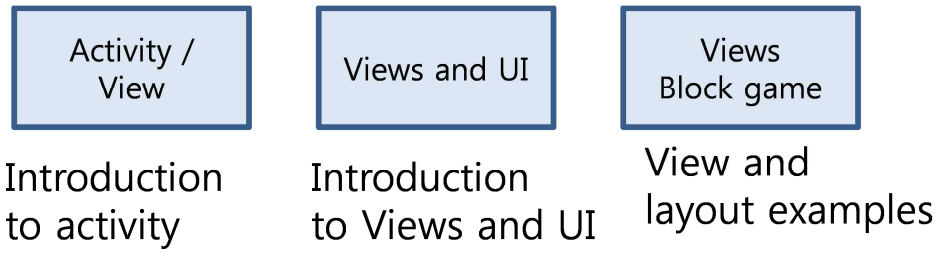


Figure 3.8. Overview of week 3

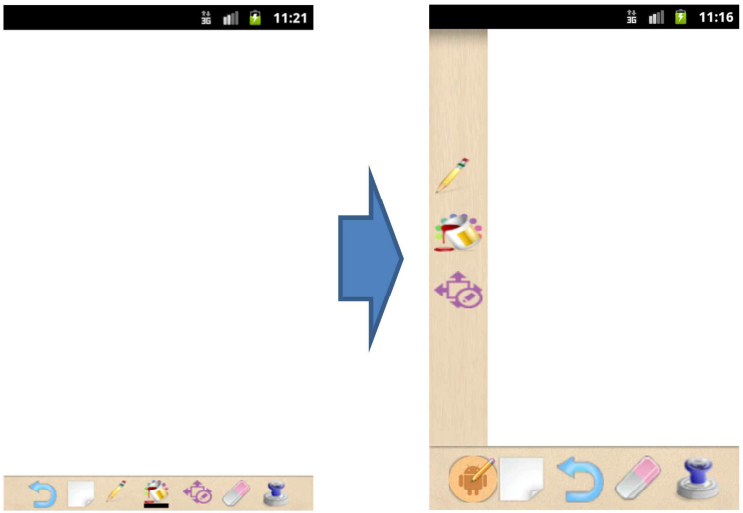
In Week 3, basic Android UI programming was covered. Basic concept of Activity, View and other UI things has been introduced. Also, advanced views such as custom view, SurfaceView, AdapterView, etc. have been explained.

Table 3.3 Overview of week 3

Monday	Activity, View – structure of activity, kind of view and view group
Wednesday	Views and UI Event Handler, MapView, AdapterView, CustomView, SurfaceView, menu, etc.
Friday	Layout change, Address book, Block game, Animation example
Lab hours	Homework of Week 2
Homework	Layout change, Address book, Block game, Animation example

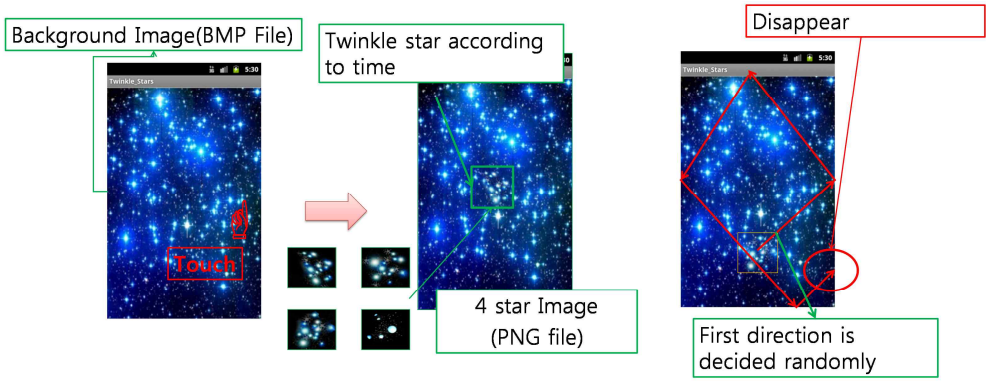
3.6.2. Exercise and homework

For the HOT, various examples were given such as layout change for the drawing App, simple address book design, touch UI change for a block game, Animation example using surface view, etc. Figure 3.9 illustrates a sample UI design instance.



(a) Layout design for a drawing app

(b) Address book : an AdapterView example



(c) Animation example using SurfaceView

Figure 3.9. Examples of week 3

3.7. Week 4

3.7.1. Overview

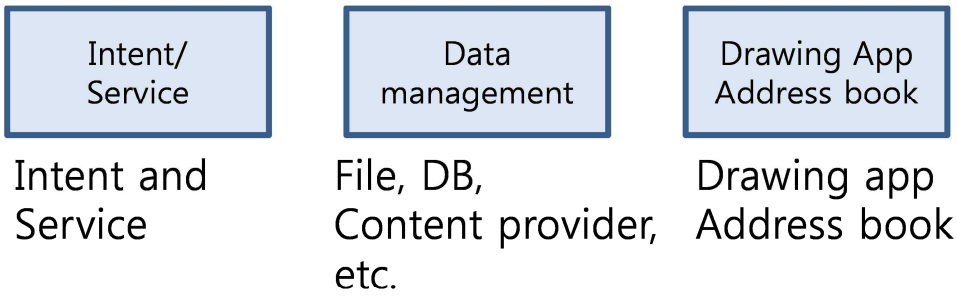


Figure 3.10. Overview of week 4

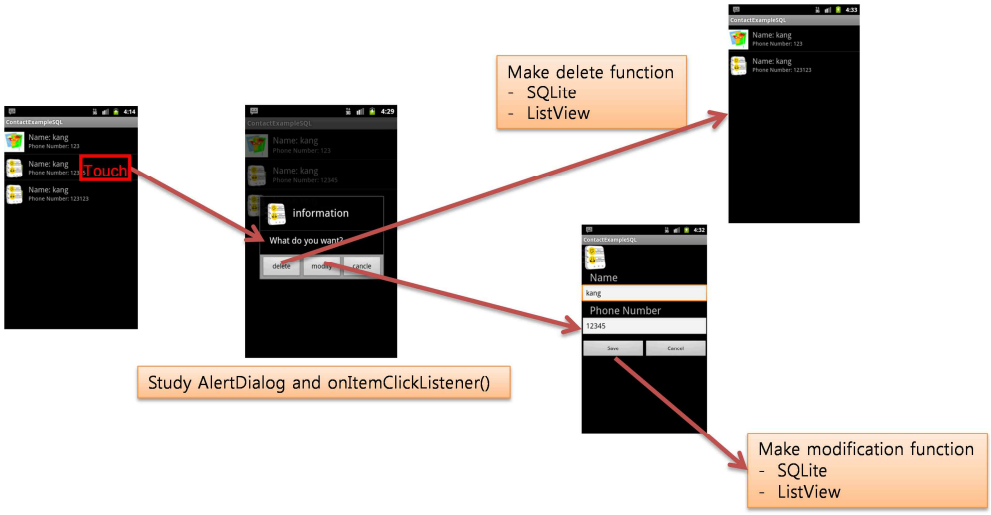
In Week 4, Intent, Service, File management, Database, etc. were covered. On Monday, we explained the concept of intent, service, broadcast receiver, etc. and On Wednesday, data management related terms were discussed.

Table 3.4 Overview of week 4

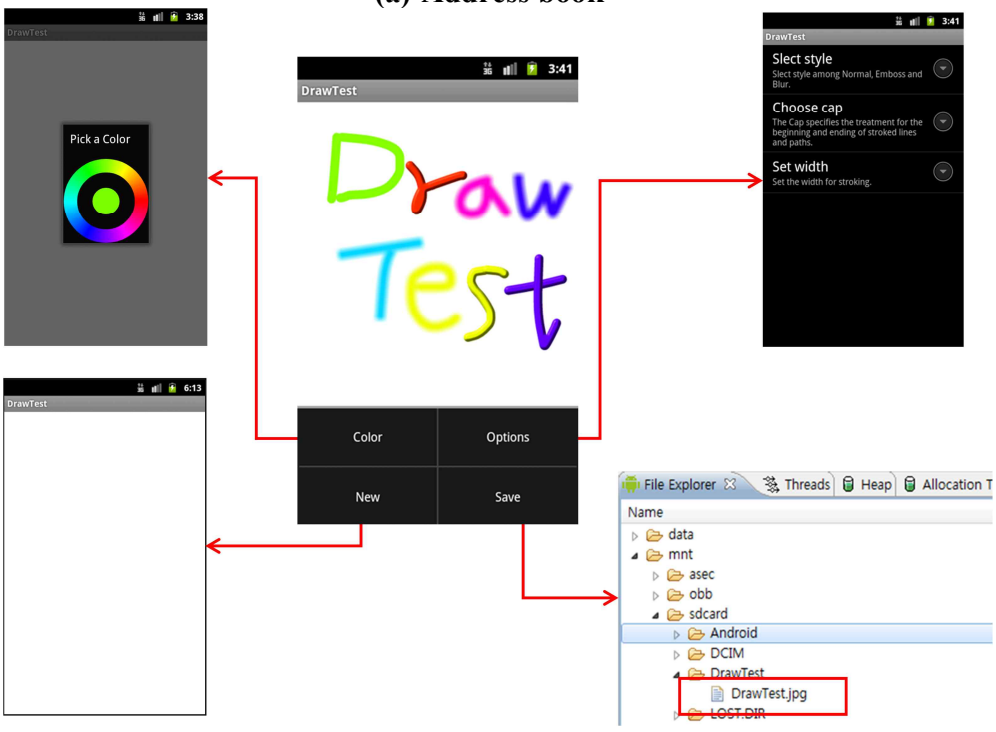
Monday	Intent, Service – Intent, broadcast receiver – Local Service, Remote Service
Wednesday	Data management – bundle, parcelable – preference, SQLite, File I/O - Content provider
Friday	Drawing App, Address book, Service example
Lab hours	Homework of Week 3
Homework	Drawing App, Address book, MapView

3.7.2. Exercise and homework

For the exercises, we presented an address book, drawing app and a simple service application which awakes an app with SMS, as shown in Figure 3.11.



(a) Address book



(b) Drawing Example

Figure 3.11. Example of week 4 apps

3.8. Week 5

3.8.1. Overview

Figure 3.12. Overview of week 5

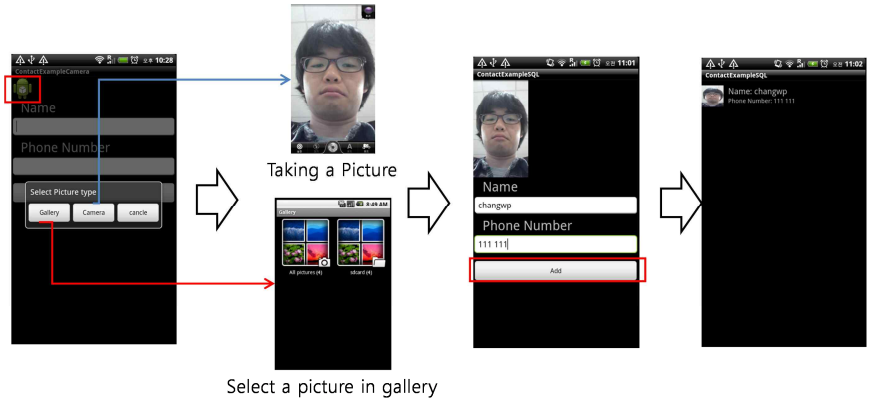
In Week 5, the theme was 'multimedia'. On Monday, related trends and basic theories were presented including audio and video signal processing. On Wednesday, Android APIs related to multimedia were introduced.

Table 3.5 Overview of week 5

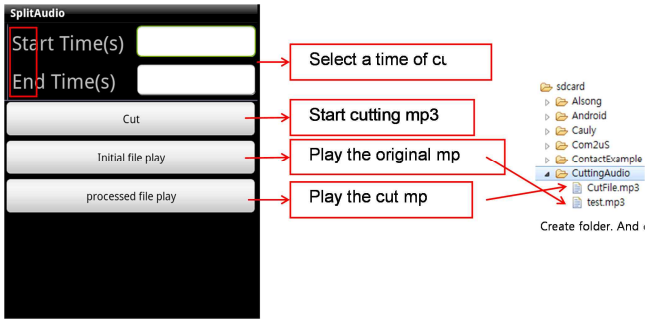
Monday	Overview – Audio and Video signal processing - Other issues in multimedia
Wednesday	Android multimedia – Media player – VideoView Camera - Etc.
Friday	Ringdroid, Cocos2D, FFMPEG, AndAR
Lab hours	Homework of Week 4
Homeworek	Address book, Audio cutting, Cocos 2D example, AndAR

3.8.2. Exercise and homework

On Friday, we analyzed and explained various open source codes and developed Apps such as FFMPEG based video cutting, Ringdroid, simple 2D game with Cocos 2D, 3D object display with AndAR, etc. (Fig. 3.13).

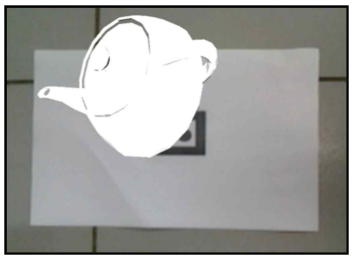


(a) Address book using camera



(b) Audio cutting using Rindroid

(c) Video cutting using FFMPEG



(d) AndAR

Figure 3.13. Example of week 5 apps

3.9. Week 6

3.9.1. Overview

Figure 3.14. Overview of week 6

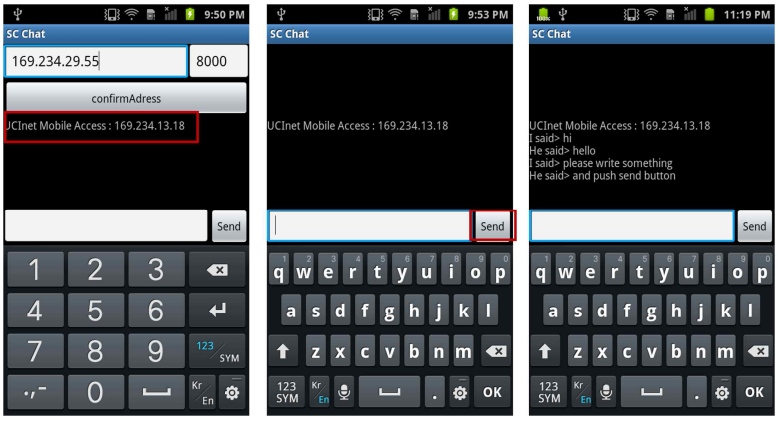
Week 6 covered network related aspects. On Monday, basic knowledge for network and communication was introduced including wireless communication, Wi-Fi, Buletooth, NFC, etc. On Wednesday, Android APIs related to network were presented.

Table 3.6 Overview of week 6

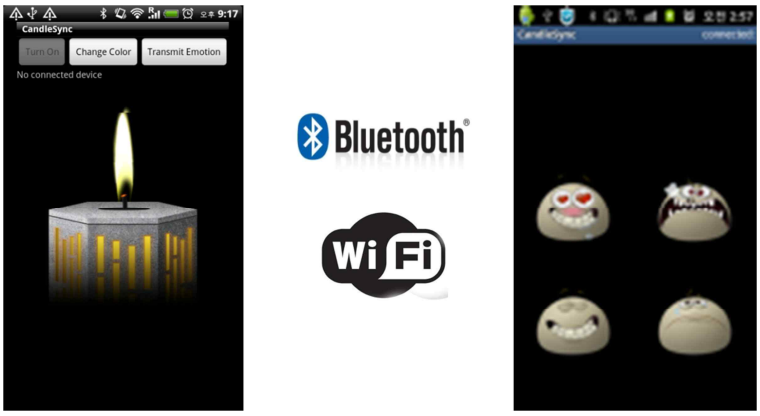
Monday	Overview – Network - Wireless communication -Bluetooth -WLAN - Other issues in network
Wednesday	Android network – 3G - WLAN –Bluetooth - Etc.
Friday	BlueChat, Wi-Fi based Chatting, Bluetooth based data exchange
Lab hours	Homework of Week 5
Homework	Wi-Fi based chatting, Data exchange using Bluetooth in Address book

3.9.2. Exercise and homework

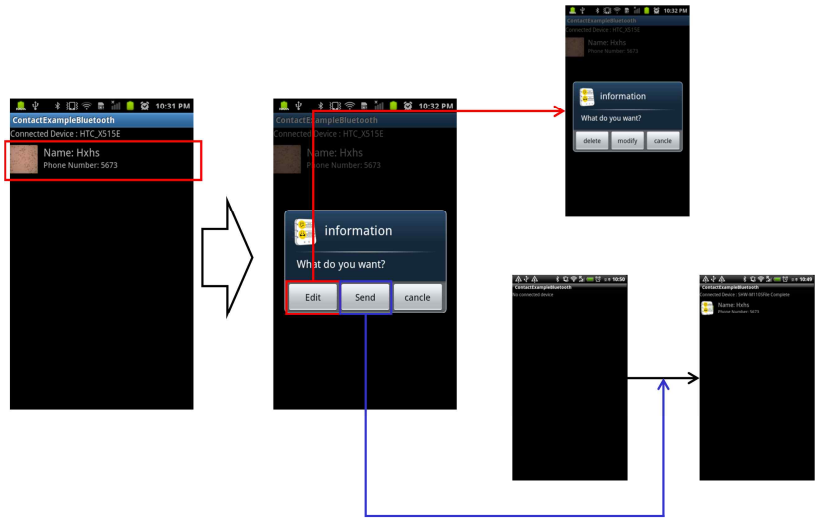
On Friday, we analyzed and explained various Apps such as Wi-Fi based chatting, CandleSync which exchanges message using Wi-Fi, Bluetooth and SMS, address book exchange with Bluetooth, etc. (Fig. 3.15).



(a) Wi-Fi based chatting



(b) CandleSync



(c) address book exchange with Bluetooth

Figure 3.15. Example of week 6 apps

3.10. Week 7

3.10.1. Overview

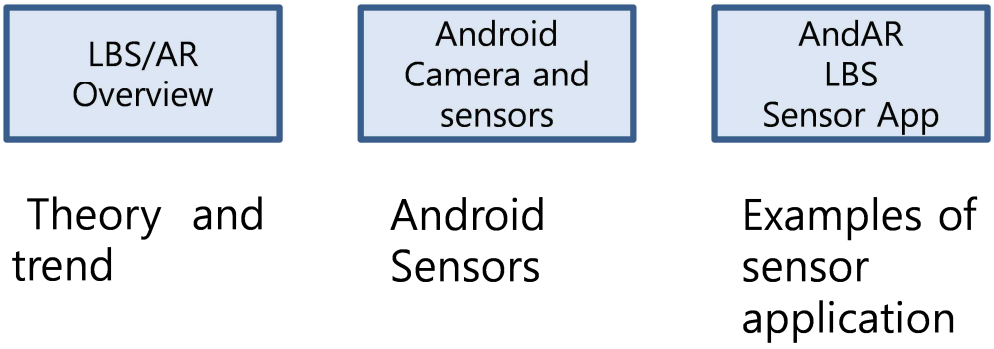


Figure 3.16. Overview of week 7

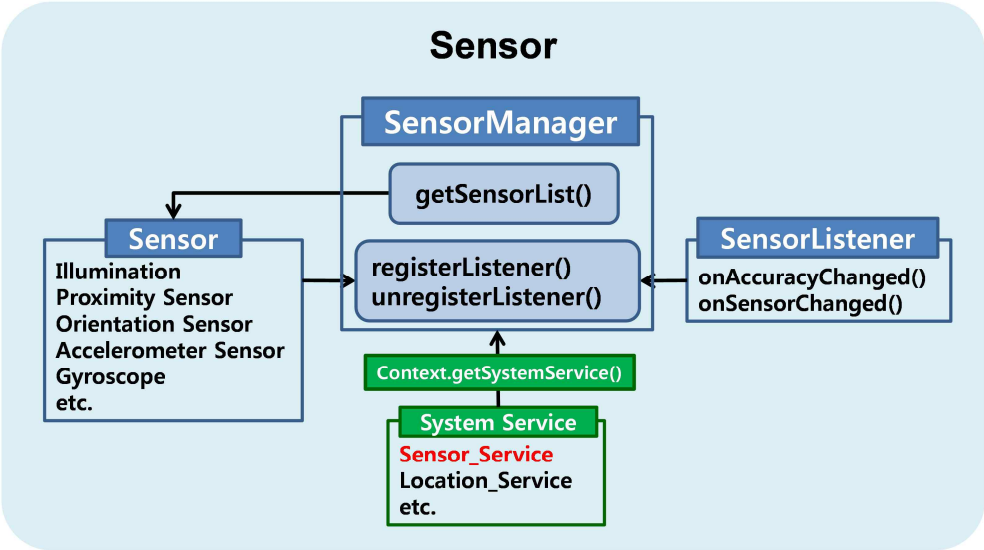
Week 7 covered sensor issues and related applications. On Monday, basic knowledge and trends about LBS [9] and AR were given. On Wednesday, Android APIs related to LBS and sensors were presented.

Table 3.7 Overview of week 6

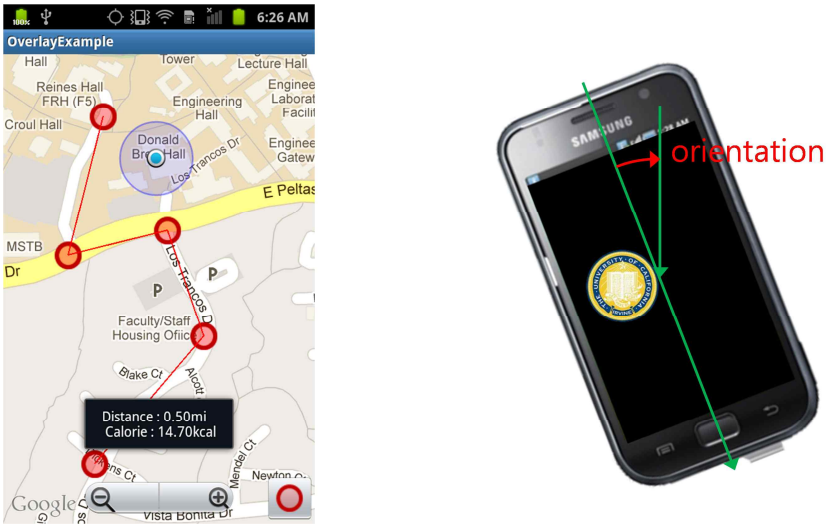
Monday	Overview - LBS - AR - Other issues for sensors
Wednesday	Android Camera Android LBS Android sensors Etc.
Friday	LBS app, Gyroscope app, Sensor based app
Lab hours	Homework of Week 6
Homework	LBS app, Gyroscope app, Sensor based app

3.10.2. Exercise and homework

On Friday, we analyzed and explained various Apps such as a simple LBS App, Gyroscope based App, Sensor based App, etc. (Fig. 3.17).



(a) Android Sensor



(b) LBS and sensor app

Figure 3.17. Example of week 7 apps

3.11. Week 8

3.11.1. Overview

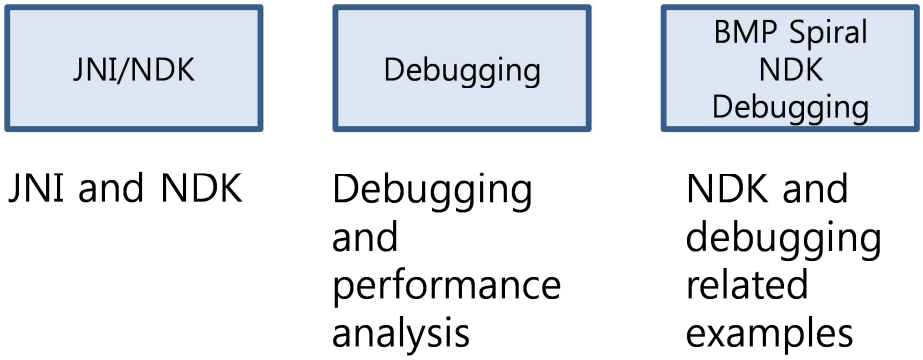


Figure 3.18. Overview of week 8

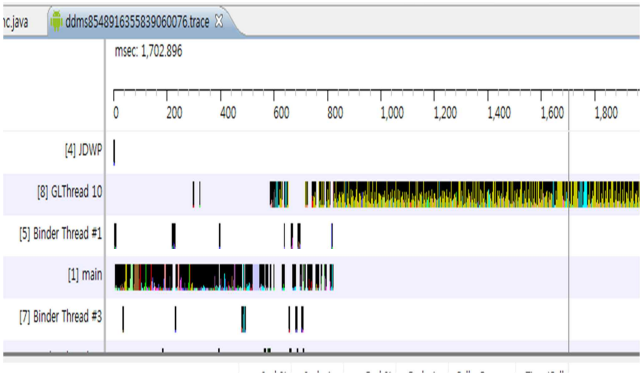
In week 8, we introduced NDK, debugging and performance analysis issues. On Monday, numerous development methodologies using NDK were introduced with examples. On Wednesday, debugging and performance analysis were discussed. DDMS, Traceview, profiling and debugging methods were explained.

Table 3.8 Overview of week 8

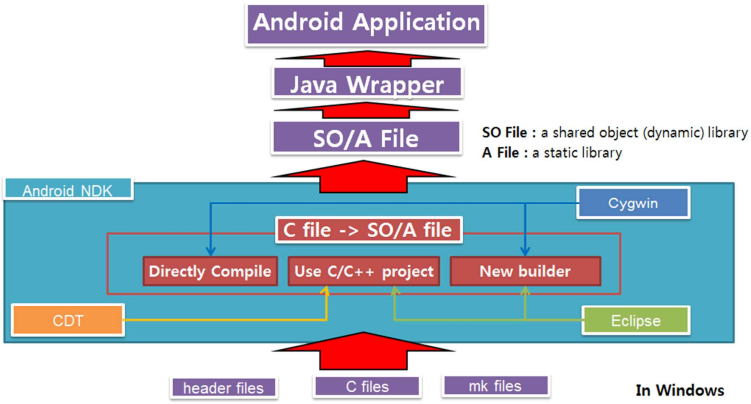
Monday	JNI/NDK - JNI - NDK - NDK development
Wednesday	Debugging and performance analysis - DDMS - Traceview - Profiling - Debugging - Etc.
Friday	Spiral image app, FMOD, etc.
Lab hours	Homework of Week 7
Homework	Spiral iamge

3.11.2. Exercise and homework

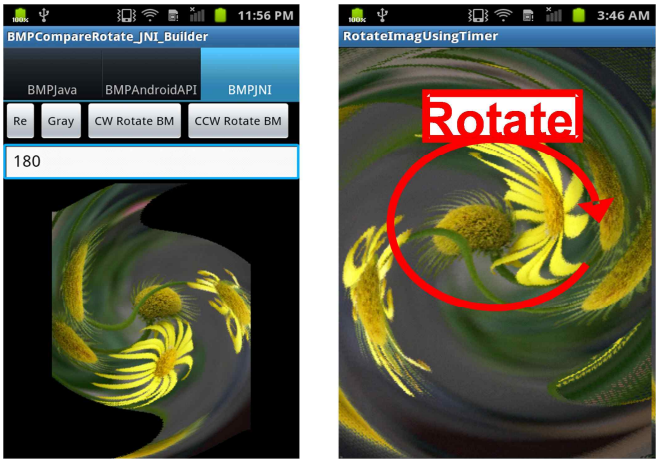
On Friday, as an exercise and homework, we dealt with a spiral image example, Wolf3D, FMOD, etc. (Fig. 3.19).



(a) Android profiling



(b) NDK development



(c) Spiral image using NDK

Figure 3.19. Example of week 8 apps

3.12. Week 9

3.12.1. Overview

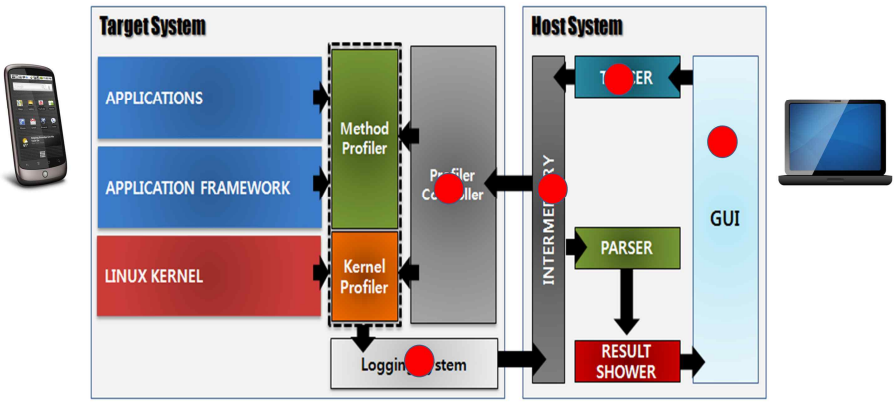
Figure 3.20. Overview of week 9

From Week 9 and Week 10, we introduced recent trends for smartphones and embedded systems. In Week 9, we discussed mobile embedded system related issues for the development of smartphone and Apps, many of which are on-going efforts in research and academia. On Monday, we introduced power and energy issues in smartphones [10][11]. On Wednesday, performance analysis and optimization issues were introduced. An example performance analysis tool was introduced and optimization issues [12] were discussed.

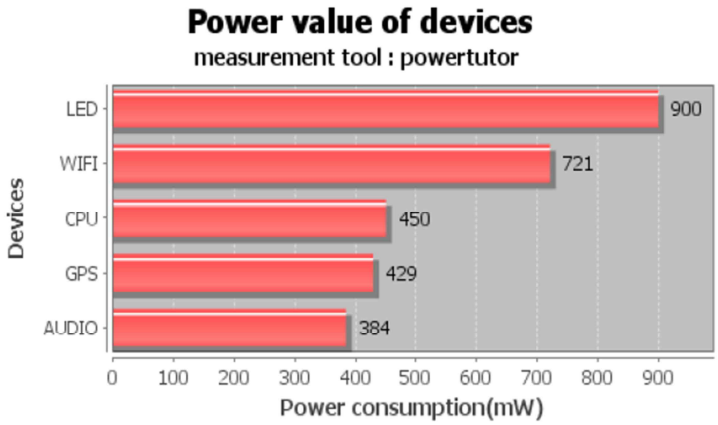
On Friday, we dealt with cloud computing issues. We introduced the basic concept of cloud computing, related applications, and the summary of OpenStack.

Table 3.9 Overview of week 9

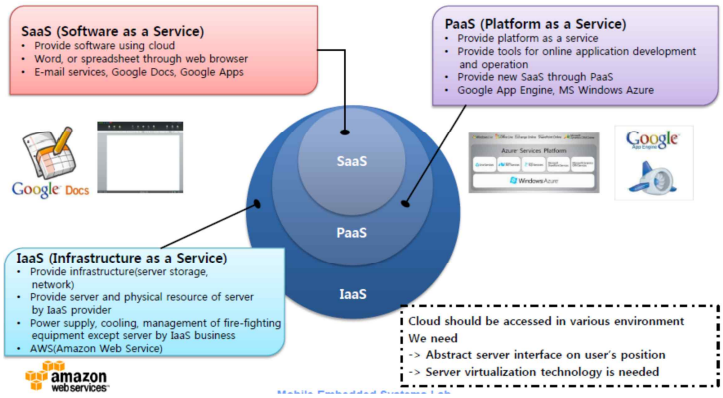
Monday	Power dissipation and energy
Wednesday	Performance analysis and tools Optimization
Friday	Cloud computing
Lab hours	Homework of Week 8
Homework	No homework



(a) Performance analysis tool in [10]



(b) Power consumption example [11]



(c) Cloud computing example

Figure 3.21. Examples of week 9

3.13. Week 10

3.13.1. Overview

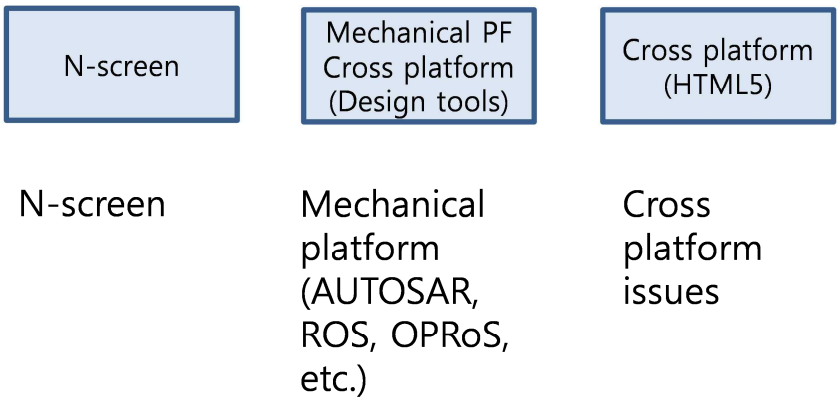


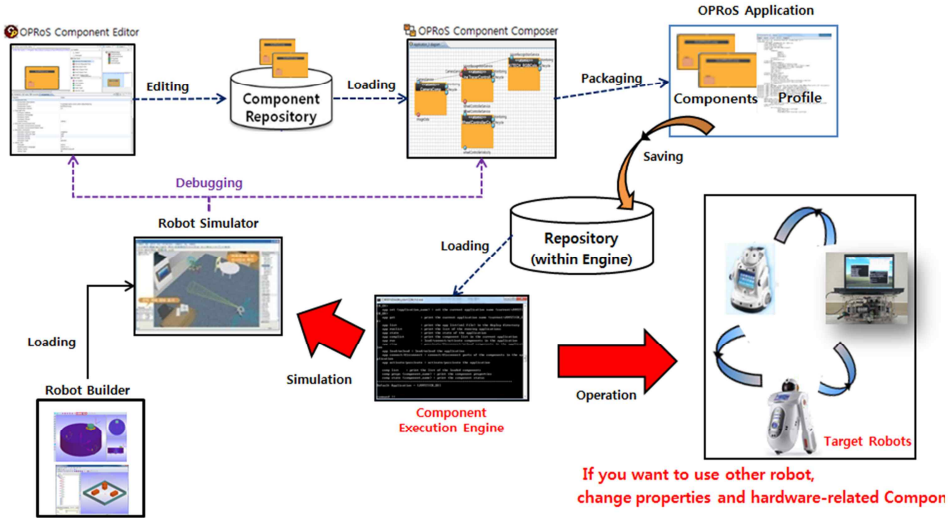
Figure 3.22. Overview of week

In the last week, other recent topics were introduced. N-screen is a new keyword relating to industry convergence; cross platform issues becomes more and more important nowadays for the development of applications. We introduced the trend of N-screen, robotic platforms including ROS and OPRoS [13], and the trend of cross platform including Verivo, HTML5, Adobe PhoneGap, Mozilla B2G and WAC (Wholesale Application Community)

Table 3.10 Overview of week 10

Monday	N-screen issue
Wednesday	Mechanical platform - AUTOSAR - ROS - OPRoS
Friday	Cloud computing
Lab hours	Project related issues
Homework	No homework

(a) N-screen & Cloud service



(b) Robot application development using OPRoS

(c) Comparison of HTML 5, B2G and PhoneGap approaches for the cross platform

Figure 3.23. Examples of week 10

4. Project

4.1. Project schedule

As described earlier, a major outcome of the course was a final project developed by teams of 2-3 students. A total of 13 teams completed the course projects and participated in the AMASE [7] project showcasing event.

Table 3 shows the detailed schedule for the team projects, that progressed in parallel with the lectures and homework/HOT sessions.

Table 4.1. Schedule for the term projects in CS 190

	What to do	Due date
Team	Making teams(2-3 students per team)	Week 1-3
Project plan	Detailed project plan Final project plan (Week 5)	Week 3-5
Development & discussion	Weekly reports	Week 3-10
Final report and presentation	Final reports-paper and app report (Week 10) Final presentation (Week 11)	Week 10-11
Showcase	Showcase and presentation	Week 11

Templates for the project plan, paper report paper and app report have been given in Appendix.

4.2. Tentative projects

Tentative projects have been recommended by the instructors which can be easily made using the provided examples.

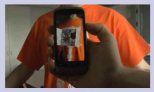
The followings are the given examples:

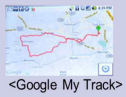
- Remote control of robots using smartphone
- Cell-bot / Cloud robotics
- Black box application in automobile
- Character recognition using smartphone
- AR(Augmented Reality) application using NYAR or AndAR tool kit
- LBS application in UCI
- LBS application in UCI using AR
- Mobile album – connect UCI
- Send my firework !
- App analyzer
- Control My PC

Figure 4.1 shows some examples of the sample projects. Some of given projects have been selected by the teams. The other teams derive their own subjects.

Project	Remote control of robots using smartphone
Examples	 <Kookmin Univ.>  <Keio Univ.>  <Univ. of West of Eng.>
Description	Control the robot using smartphone Bluetooth / Wi-Fi / WCDMA connection N-screen between robots and smartphones E-learning using robot for children OPRoS / ROS platform
Sample Applications	iRobiQ, KMU/Yujin robot Hexapod Robot, University of the West of England Walky, Keio Univ.

Project	Black box application for automobiles
Examples	 <DailyRoads Voyager >  <Speedmate blackbox, SK>
Description	Record audio and video for automobiles Detection of status such as emergency Video codecs such as FFMPEG can be utilized for higher level editing Various functions can be added
Sample Applications	DailyRoads Voyager Speedmate blackbox, SK

Project	Augmented Reality using AR toolkits
Examples	  <Popcode, U. Cambridge>
Description	AR applications using AR toolkits NyAR, AndAR tool kits On/off-line applications using AR engine and OpenGL
Sample Applications	NyAR AndAR Popcode Vuforia

Project	Find my room in UCI (LBS applications)
Examples	  <Bionic Eye>  <Ovjet>  <Google My Track>
Description	LBS applications AR tool kits can be used Find the direction & distance
Sample Applications	Bionic eye Google My Tracks Ovjet

Project	Cell-bot / Cloud robotics
Examples	 <UCI>  <KMU>   <Google cellbot / CR >
Description	Autonomous control of robots using smartphone Smartphone plays a role in main processor for robot Make smarter robot using smartphone or cloud robotics
Sample Applications	Android car, UCI X-bot, KMU/Yujin Robot Goggles, Google Cellbot / Cloud robotics, Google Roomba, i-robot

Project	Mobile Album - Connect UCI !
Examples	
Description	SNS Collect photos and addresses of UCI MEMBERs
Sample Applications	SNS apps Bump Hoccer

Figure 4.1. Examples of sample projects given by instructors

4.3. Project results

Table 4 shows the term projects of the students. Figure 6 shows the examples in CS 190 projects.

Finally, in Week 11 (finals week), we held the AMASE showcase [7] with the sponsorship of many companies including SK Planet, Samsung, Google, KITECH and D&CI. Over 100 attendees participated in the AMASE event and we also had news coverage by the local media.

Table 4.2. Term projects in CS 190

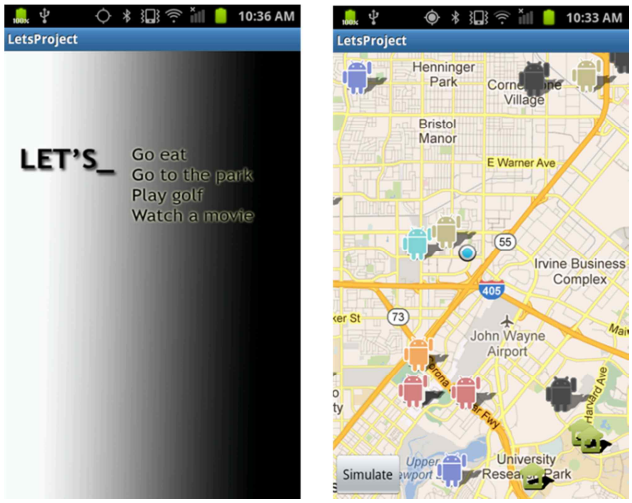
Team name	Theme	Categories
Let's	Location-based social networking service	LBS, SNS
Wifi Flash drive	File sharing service between smartphone, PC and servers	Network, File sharing
RoboKontrol	Interaction app with OPRoS based robot	Remote control, Robot Gesture Recognition
Scheduleshare	Schedule sharing between users with 'BUMP' APIs	Schedule sharing
Gyaan	Quiz taking app with SNS characteristics	SNS, Game, Quiz
GBC (Ginger Bread Car)	app interaction with Android car using OpenCV	Robot, Pattern recognition
ShuttleScheduler	UCI bus system app providing bus/walking navigation at UCI	LBS, Navigation
BooksInteractive	AR app for child books using Qualcomm Vuforia	AR, Learning
Dungeon Master	SNS based game which simulates combat between players and monsters	Game, SNS
PhotoGam	SNS based photo editing app	SNS, Photo
KOBOT	Remote control of an OPRoS based robot	Remote control, Robot
IFM (Intelligent Food Monitor)	A reminder app that tracks and monitors your groceries and diet	u-health, Barcode Scanner

4.4. Abstracts from paper reports for each team

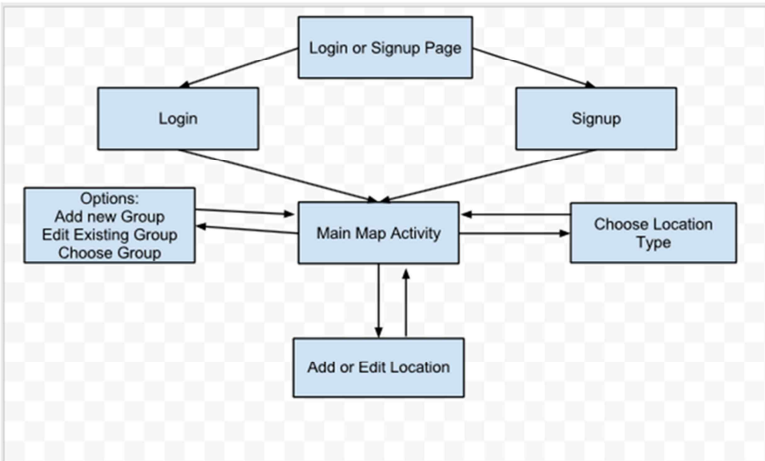
4.4.1. Let's

Let's is a location-based social networking awareness application for mobile devices, such as smart phones. It is best known for letting Smartphone users tell others about nearby friends, activities, and events. User can find nearby friends without having to search for them. The app will recommend what places you frequent or what your friends like.

Let's is a mobile application that allows registered users to post their location at a venue and connect with friends.



(a) App Screenshot

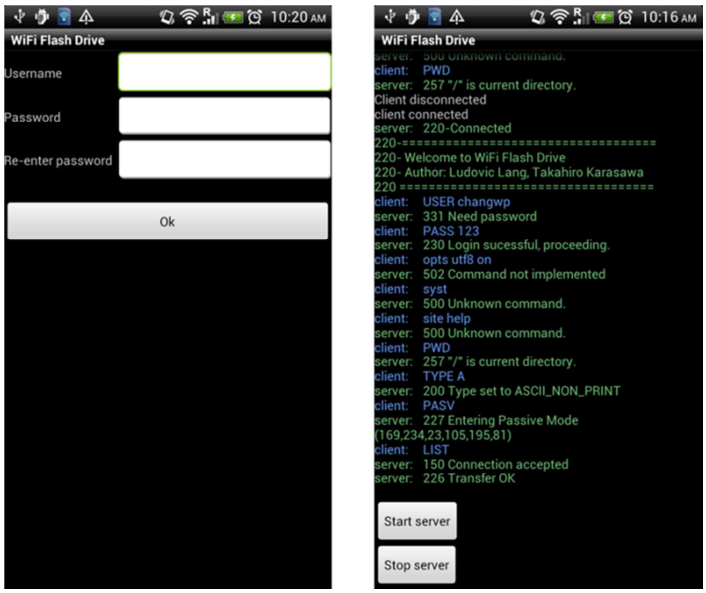


(b) Application structure

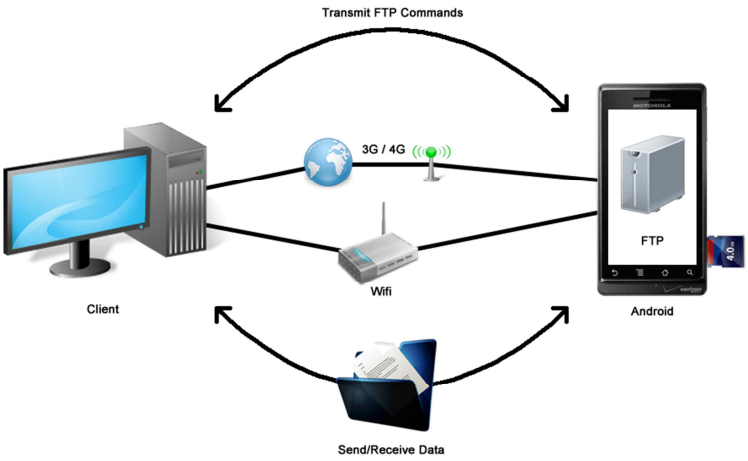
Figure 4.2 'Let's' Project

4.4.2. Wi-Fi Flash Drive

This is a file transfer app for Android, allowing computer users to read and save files to the android phone just like a flash drive but over a wireless connection (Wifi and 3G/4G). With this app, the user is not required to install any software or drivers, or bring any kind of cables to utilize the 32 GB SD card inside his/her phone as a flash drive



(a) App Screenshot

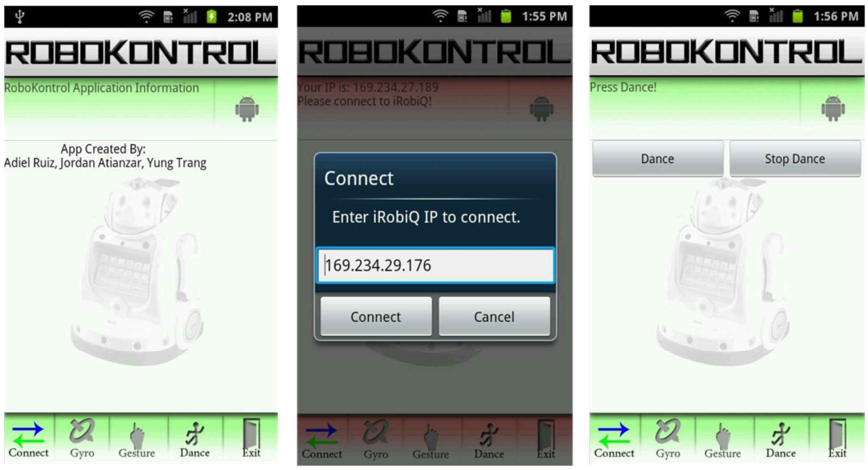


(b) Application structure

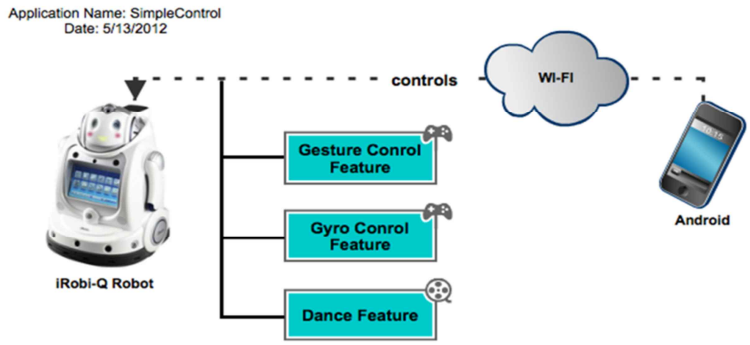
Figure 4.3 'Let's' Project

4.4.3. RoboKontrol

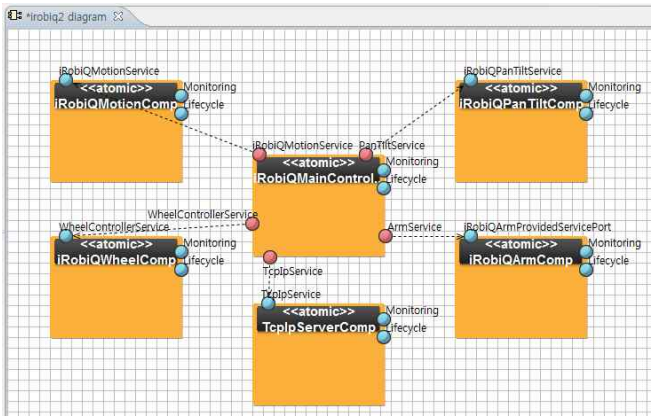
RoboKontrol is an android application that controls iRobiQ with new and friendly techniques like gesture and gyroscope. This document reports the background, components, issues, reflections, and suggests further development for the application. The main RoboKontrol components addressed in this report are the gesture control, gyro control, dance feature, network, and the GUI. After three week developing RoboKontrol application, our final result is that RoboKontrol is able to allow users to use gestures and gyro to control iRobiQ effectively. Additionally, the dance feature of this application demonstrates the potential promising connotation that our application can be expanded and growth in entertaining aspect to please the user experiences. This report also records the technical and nontechnical issues during the design, implementation and debugging process. In addition, areas of improvement are addressed for further development and maintaining purposes in the future. The three main tools we used frequently in this project are Eclipse, EGit with bitbucket, and virtual iRobiQ and Opros (later replaced with the actual iRobiQ for testing). The programming language we used for implementation is Java with Android API.



(a) App Screenshot



(b) Network Environment

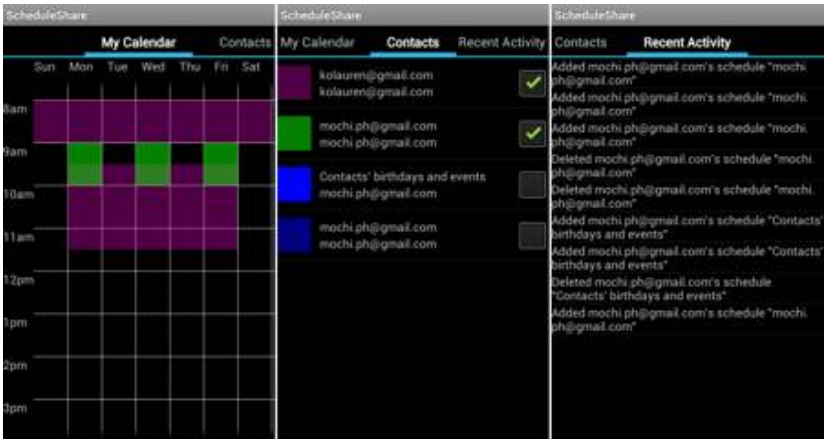


(c) OPRoS Application Diagram

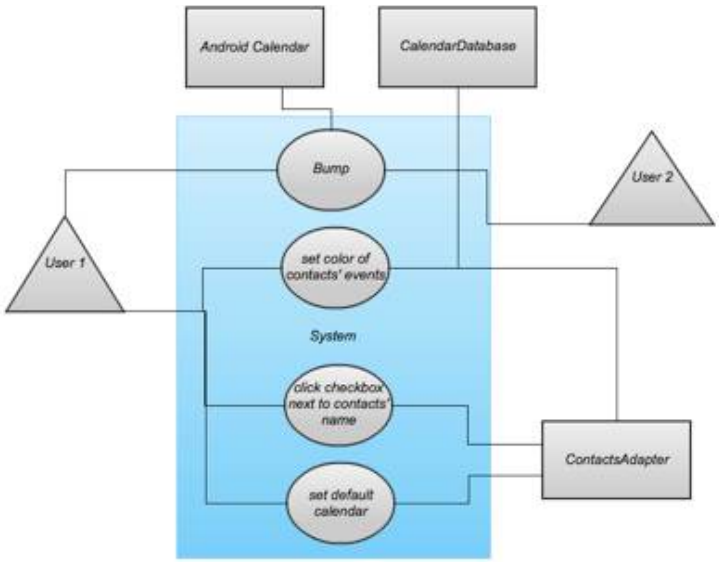
Figure 4.4 RoboKontrol

4.4.4. ScheduleShare

ScheduleShare is an application that easily trades schedules between two people by "bumping" together their phones using the Bump API (http://bu.mp/api). The app displays the calendars that are already in your phone. The user can choose which calendar to display. When they "bump" their phone with another user, their schedules will be sent each other. Users can use this app to see when their schedules overlap and schedule meetings accordingly. ScheduleShare requires Android 4.0+ since it uses the CalendarProvider API (only available in Ice Cream Sandwich).



(a) App Screenshot

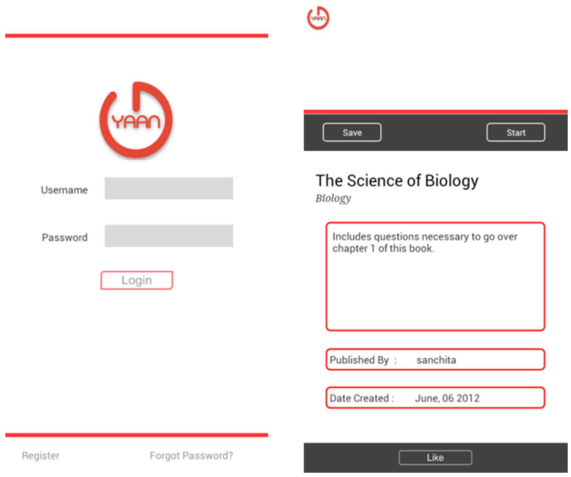


(b) Application Structure

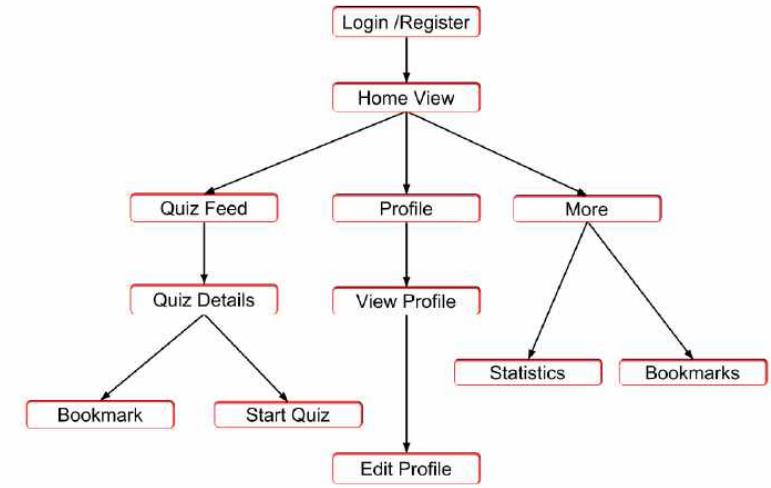
Figure 4.5 ScheduleShare

4.4.5. Gyaan

Gyaan is a quiz taking application designed to help students learn and improve their skills in subject (or topics) they choose. It allow educators to create and publish quizzes for users on the application (students). It does it by taking advantage of technologies internet, smartphones, laptops and tablets. The main purpose of the service is to make learning easy without compromising the social aspect of internet.



(a) App Screenshot



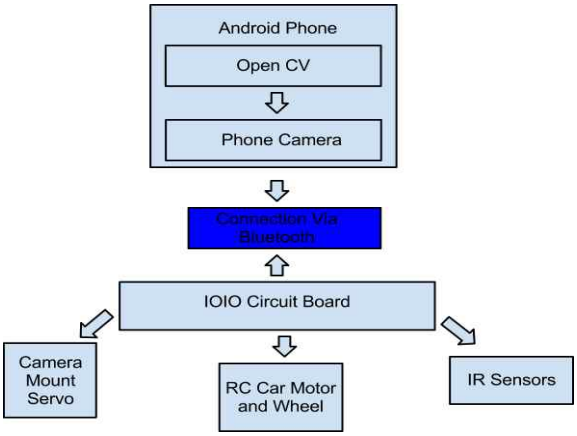
(b) Application Structure

Figure 4.6 Gyaan

4.4.6. GBC

GBC is a complete remodel of the old Android Car that was released. Version 2.0 operates solely through calculations on the android phone, noting that we are still using the hardware of the first version using the IOIO board to integrate the RC car with the android platform. There are two main components to the new Android Car. The first introduces object tracking using OpenCV libraries so that the RC car can find and track objects using the android phone camera. The other main component is the ability to navigate a maze using our own algorithm that combines traveled distance using a hall sensor to the RC car, with error correction using trackable object landmarks to allow the RC car to effectively navigate through the end of a maze.

(a) Autonomous Vehicle

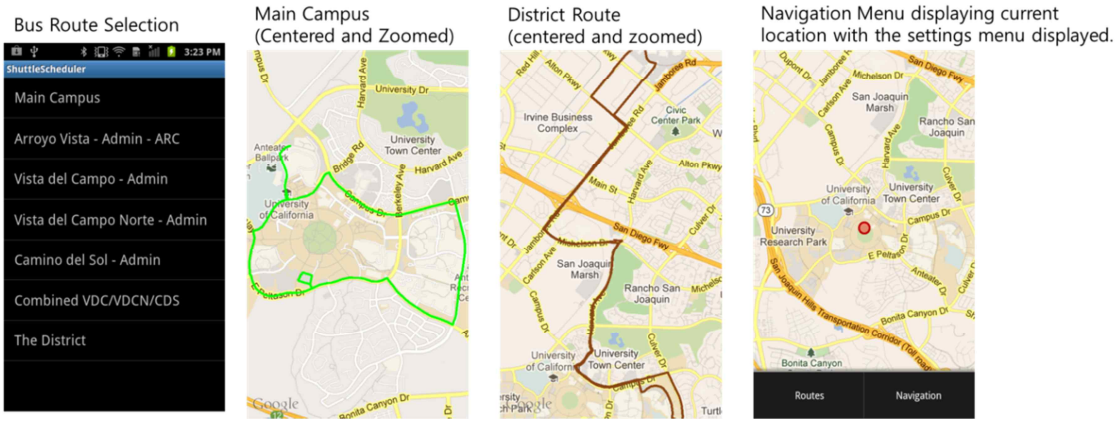


(b) Application Structure

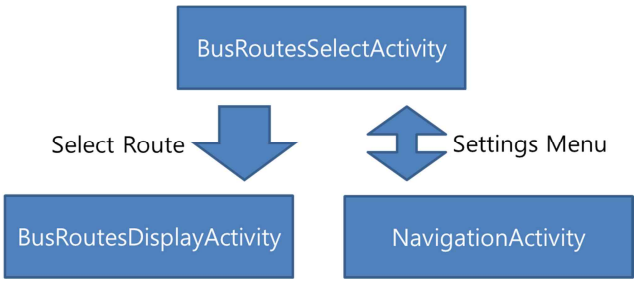
Figure 4.7 GBC

4.4.7. ShuttleScheduler

To facilitate the use of the Anteater Express shuttle system, we have developed a mobile application that utilizes global positioning system (GPS) technologies, internet connectivity, and the Google Maps libraries found in Android smartphones to assist students with scheduling trips around the University of California, Irvine (UCI) campus. This will allow students to maximize the use of their university-run public transportation system by allowing them to more effortlessly navigate throughout campus and to abstract away the complicated logistics involved with reading and planning trips based on the myriad of schedules that exist for the Anteater Express.



(c) App Screenshot



(d) Application Structure

Figure 4.8 ShuttleScheduler

4.4.8. BooksInteractive

BooksInteractive is a mobile application developed with the intent to explore the up and coming concept of augmented reality on the smartphone and its potential to be used as a learning aid. The application's functionalities include augmenting images in children's books through the camera lens of a mobile device, accessing multiple augmented books in memory, and downloading augmented book content from the internet. The content of the report is made up of design decision made and issues faced in the development of the application as well as potential improvement that can be made.

Figure 4.9 BooksInteractive

4.4.9. Dungeon Master

Dungeon Fighter is a game which simulates combat between players and monsters. It is similar to playing Rock, Paper, Scissors, since there are three moves a player can choose when fighting (attack, reload ammunition, defend). However, Dungeon Fighter is a more complex game as players must be careful as to not be caught off-guard without ammo and they must defend their health points. Players can either play in Story mode or Bluetooth mode. Story mode is a single-player mode in which the Player plays against increasingly harder enemies with more life points. Bluetooth mode is a two-player mode in which two players can connect through each other with Bluetooth and fight against each other.

Figure 4.10 Dungeon Master

4.4.10. PhotoGam

The PhotoGam application attempts to meet the need for sharing content from multiple sources to multiple sources. Oftentimes in competing applications, they focus on sending content from multiple sources to their service; however, our application focuses solely on providing the content sharing platform so that in our case, images could be sent from a variety of sources and sent to a variety of services and other platforms.

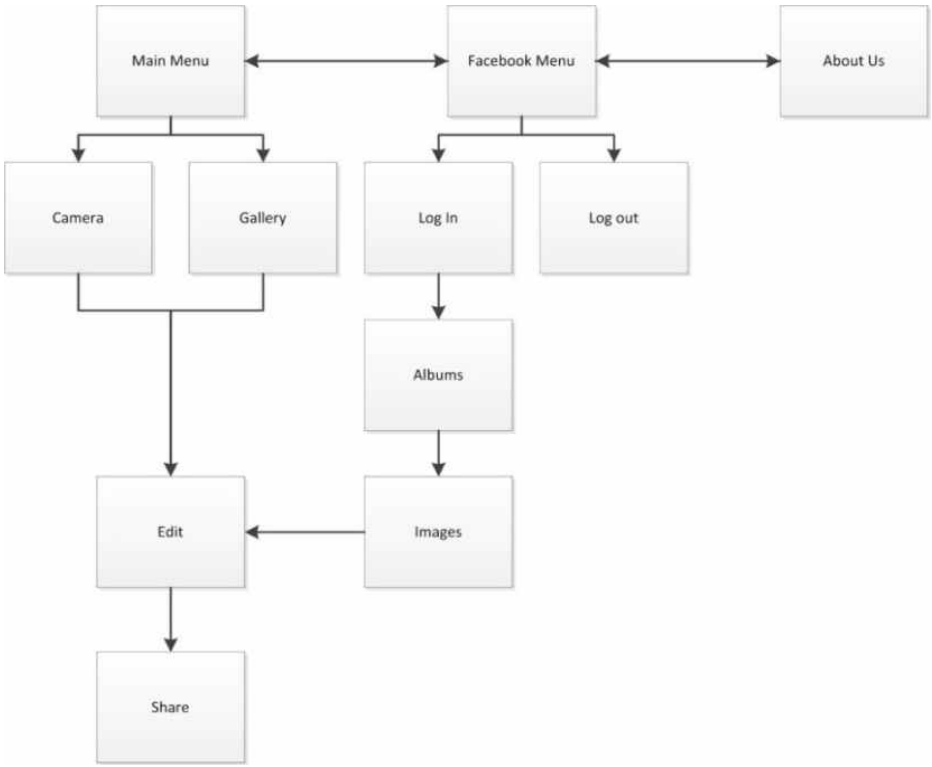
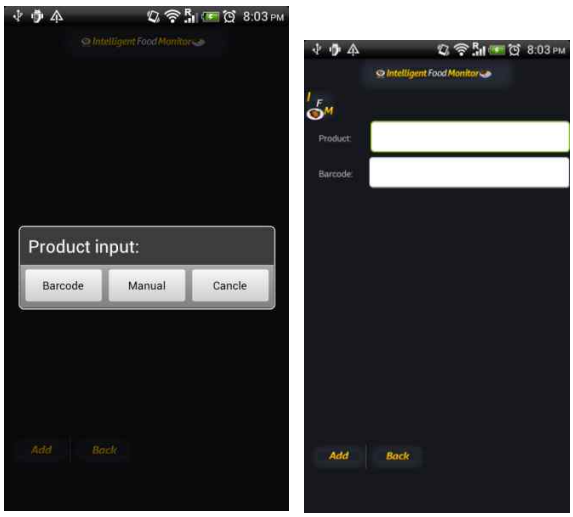


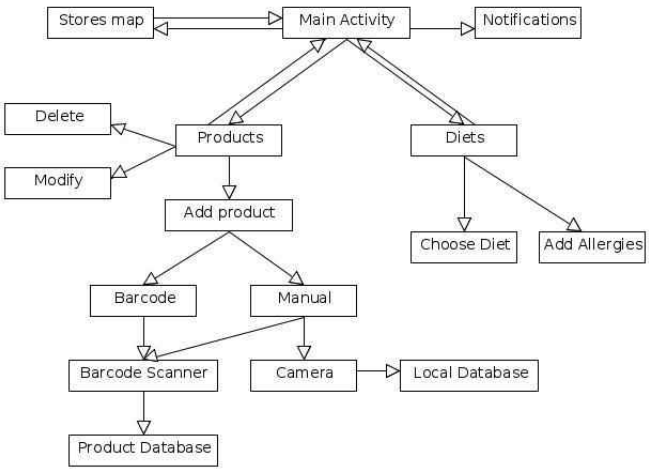
Figure 4.11 PhotoGam

4.4.11. IFM(Intelligent Food Monitor)

This document describes our android Intelligent Food Monitor Mobile app in short the IFM. The Intelligent Food Monitor Mobile app was designed as a reminder app that tracks and monitors your groceries and diet. The development of modern smart phones has given us the ability to integrate technology to our daily lives. It gives us access to the power of the web in the palm of our hands. Using this advantage we want to design a smart intelligent Food monitoring system that can be integrate in to a person's live effortlessly. It keeps track of our diet and groceries. By tracking a person's food in takes and available food at home the system calculates and notifies the user when they are running out of a certain food item and when to go to the store to buy groceries. The system also able to track expiration dates of foods.



(a) App Screenshot

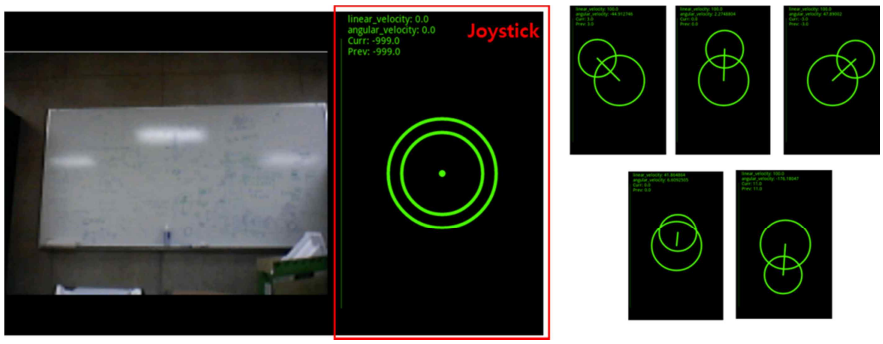


(b) Application Structure

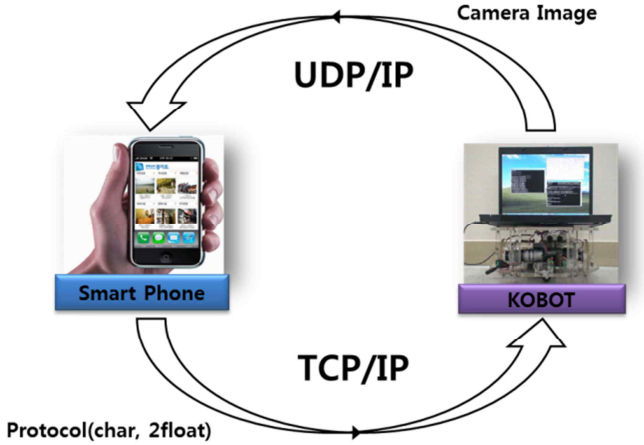
Figure 4.12 IFM(Intelligent Food Monitor)

4.4.12. KOBOT

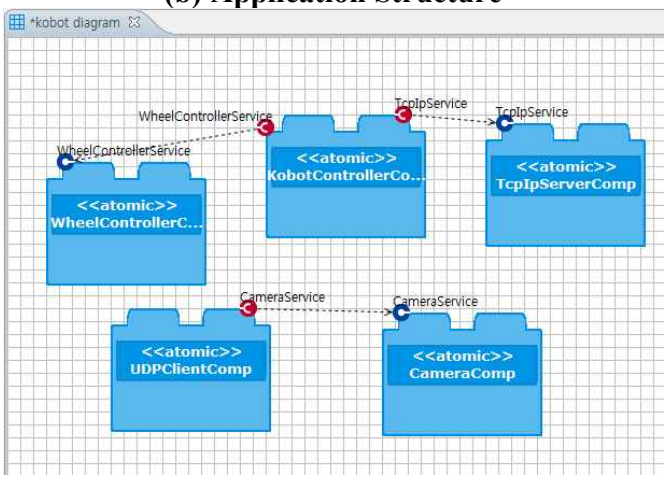
Kobot is a robot powered by the OPRoS platform. Communication between an Android phone and a laptop running the OPRoS Engine controls Kobot and allows a video feed to be sent back to the user so that the user can see what Kobot sees.



(a) Android App Screenshot



(b) Application Structure



(c) OPRoS Application Diagram

Figure 4.12 KOBOT

5. Evaluation of the course

Table 2 shows the rubric for evaluation in CS 190 lecture. Our goal was to ensure that students were not only comfortable in Android programming but also in all aspects of embedded software development and deployment, including project management, feasibility and utility of project, and usability of the software. Accordingly, we divide evaluation into two parts, homework and project with 40 and 60 points, respectively.

Table 5.1. Course evaluation rubric for CS 190

(a) Overall score

	Score
Homework	40
Project	60
Total	100

(b) Project rubrics

	Percentage	**Score**
Creativity	20	12
Market	30	18
Technology	30	18
Usability	20	12
Total	100	60

6. Discussion

6.1. Student feedback

6.1.1. Overview

We received excellent feedback from the students in CS 190, which is gathered from students' final reports and lecture management system. Although there was a mix of students - ranging from freshmen to seniors, and also in expertise, from novel to expert Android programmers - we received an overwhelmingly positive evaluation from the course participants. Students acknowledged that the course was challenging in their demands for time, but every single student appreciated and thoroughly enjoyed the course offering.

6.1.2. Future improvements to the course

The most common concern, if any, from the students was that the homework and assignments took a tremendous amount of time. Indeed some students indicated that the homework and lab assignments took 2-3 times more than other courses.

On reflecting back as to why the students felt this way, we think several issues contributed to this situation. First, unfortunately the course was not assigned an official Teaching Assistant (TA), who could lead lab discussions and hold additional office hours. Second, we did not allocate specific/mandatory "lab" hours where students could interact with the TAs and solve the homework and lab assignments. Although the students were given many sample source codes, they did not get an opportunity to review these samples with a TA in a lab setting. We believe the addition of a lab section run by a TA would resolve this issue.

6.2. Toward future lectures

6.2.1. Suggestion on the future course design

Figure 5.1 shows the suggested course design for the future lectures. We think that if the course can be modified just a little, the burden for the course can be cut down drastically. As can be seen in Figure 6.1, if we assign mandatory lab hours, the Friday lectures can be moved into the lab hours, and there will be sufficient time for hands-on explanation by the TAs.

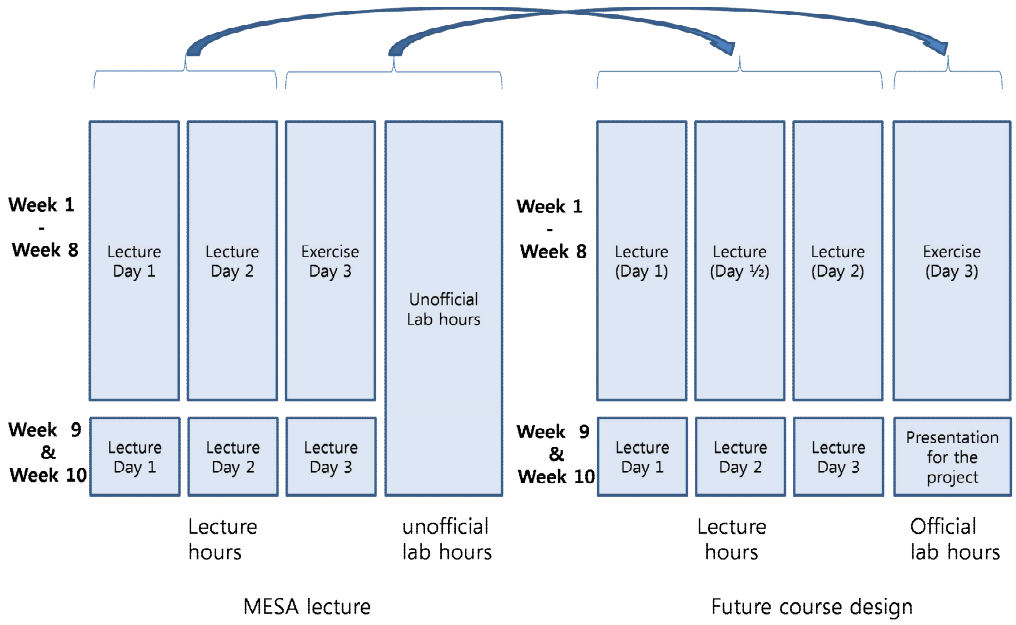


Figure 6.1. Suggested course design for the future lectures

6.2.2. What additional material could be included for future lectures ?

To keep up with the changing trends of industries and enhance the results of the students, it will be helpful to consider the following in future offerings of the course:

- More explanation for the Linux and Android platform itself
- New open platforms and API sets such as Vuforia, Samsung Galaxy Note APIs, SNS APIs, etc.
- More examples for the cross-platform issues(HTML5, PhoneGap, B2G, etc.)
- More examples for advanced apps
- Examples for mechanical platforms (ROS, OPRoS, etc.)

7. Conclusion

MESA aims to expose students to the trends of embedded systems, various devices and recent applications. There has been an increasing demand of the students for the smartphone programming lectures. However, since resources such as time, lecture support, etc. are limited, it is difficult to insert many mobile embedded software courses into the regular curriculum. Also, thanks to numerous open lecture notes and open source archives from Google, Apple and other related companies, there are already many students who have enough knowledge to develop apps.

With these goals and constraints, we developed a 1-quarter course for smartphone programming. We primarily focused on 1) introducing as many examples as possible to enhance the quality of final applications, 2) connecting other courses to smartphone programming such as embedded systems, multimedia, network, database, operating system, communication, etc., 3) analyzing famous open source projects to follow up the recent trends, 4) introducing up-to-date keywords related to the industries.

The resulting course "CS 190: Mobile Embedded SW with Android" was successfully offered at UC Irvine in Spring Quarter, 2012. The student's team projects were showcased in a highly visible public event AMASE [7] that was sponsored by several companies.

We believe that by using mobile embedded software programming on smartphones, we were able to stimulate and generate excitement in our students for embedded systems, motivating them to study other embedded system topics in the future. Given the increasingly limited resources of universities, we think this model of a course affords a good pedagogical path to reinforce and stimulate interest in an Embedded Systems curriculum.

For future offerings, we intend to insert an introduction of Android platform itself, describe more examples from newly developed open projects, and investigate cross platform related projects and mechanical platforms.

We hope this technical report will be helpful to the students who want to study mobile embedded software programming as well as the instructors who want to design a new course for the mobile embedded software programming.

8. Acknowledgment

We appreciate the support of Dr. Nicolas Oros, Mr. Chang-Woo Park, Mr. Yoon-Won Lee, Mr. Phil Ma, Mr. Kyungsu Lee, Ms. Inae Kwak, and Mr. Truong Huu Phuc for CS 190 lecture.

We also thank SK Planet, Google, KITECH, D&CI, and Samsung Electronics for their sponsorship of AMASE 2012@UCI and CS 190 lecture. In particular, we appreciate the help of SK Planet T Academy for allowing the use of 'Basic Mobile Programming' [14].

Prof. Gu-Min Jeong's effort for this course was partially supported by the MKE (The Ministry of Knowledge Economy), Korea, under the ITRC (Information Technology Research Center) support program supervised by the NIPA (National IT Industry Promotion Agency) (NIPA-2012-C1090-1221-0005).

9. References

[1] Patterson, D. A. Computer science education in the 21st century, *Communications of the ACM,* 2006, 27-30.

[2] Marwedel, P. *Embedded System Design*, Second Edition, Springer, 2011.

[3] Vahid, F. and Givargis, T. *Embedded System Design: A Unified Hardware/Software Introduction*, John Wiley & Sons, 2002.

[4] Pasricha, S and Dutt, N. *On-Chip Communication Architectures: System on Chip Interconnect,* Morgan Kaufmann, 2008.

[5] Ahonen, T. A. Full analysis of iPhone economics, http://communities-dominate.blogs.com/brands/2010/06/full-analysis-of-iphone-economics-its-bad-news-and-then-it-gets-worse.html

[6] Muppala, J. K. Teaching embedded software concepts using Android, *Proc. 2011 Workshop on Embedded Systems Education (WESE 2011),* 2011, 32-37.

[7] Official site for Android developer, http://developer.android.com/index.html

[8] AMASE 2012@UCI, http://www.ics.uci.edu/community/events/amase/

[9] Jeong, G.-M. et al. Paradigm shift and the state of the art of LBS in the advent of smartphone, *Communications in Computer and Information Science*, 2011, v. 223, 251-258.

[10] Kyong, J., Lee, M., Jeong, G.-M., Park, C., and Lim, S.-S. A systematic debugging and performance analysis framework for Android platforms, *Proc. International Workshop on Performance, Applications, and Parallelism for Android and HTML5 (PAPAH 2012),* 2012.

[11] Kim, H. Smart phone, where does the power go ? *2011 EU - Korea Conference on Science and Technology (EKC 2011),* 2011.

[12] Dutt, N. Integrating End-to-end and cross-layer optimizations for cyber-physical systems, *ARTIST Summer School in Europe 2010,* 2010

[13] OPRoS, http://210.115.36.127/doku.php?id=home

[14] Jeong, G.-M. *Basic Mobile Programming (in Korean)*, SK Planet T Academy, 2010.

10. Appendix

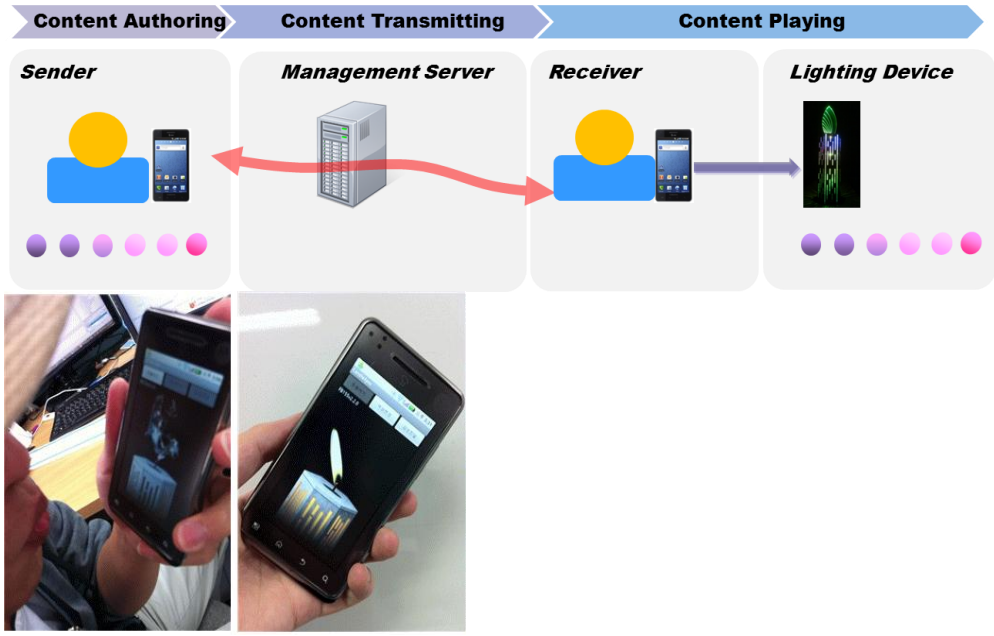
10.1. Template for the project plan

10.2. Template for the weekly report

10.3. Template for the app report

10.4. Template for the final paper

Team project plan for CS 190

Team	Candle Sync
Members	Dong-Byeong Kang Chang-Woo Park
Application name	Candle Sync
Abstract	Candle sync is an application for LED mood lighting system. Users can send their feeling or mood through the wireless network using their smartphone. User can select a color of the candle and can send it to other user's phone. When receiving message, candle sync is awaked and display it on the screen. Also, it can be connected to real LED devices through Bluetooth.
Features	Users can send their feeling or mood through the wireless network using their smartphone. User can select a color of the candle and can send it to other user's phone. When receiving message, candle sync is awaked and display it on the screen.
Reference Apps	LED
Comparison to the reference apps	Similar to LED application, user can send the content to other's phone in the aspect of LED like display. But, in candle sync, we want to emphasize 'candle' and connect to real LED devices.
Figures, block diagrams or sketches	

Categories of App (You can have multiple checks.)	Academic aspect	Technology oriented	O
		Showcase oriented	O
		Public usage oriented	
		App store oriented	
		Etc.	
	Market aspect	Campus	
		On the go	
		Entertainment	O
		Social	O
		Game	
		Lifestyle	
		Multimedia	
		Etc.	
Self-evaluation	Level 1(Very low) – 5(Very high)	Creativity	4
		Technology	4
		Usability	3
		Market	3
		Etc.	
	Technology 0 : never used 1(Very low) – 5(Very high)	User interface	3
		Graphic design	3
		Multimedia	3
		Network	4
		Connection to social network	3
		Location	0
		Camera	0
		Augmented Reality	0
		Sensor	3
		NDK	0
		Server	0
		Connection to other devices	4
		Additional hardware design	4

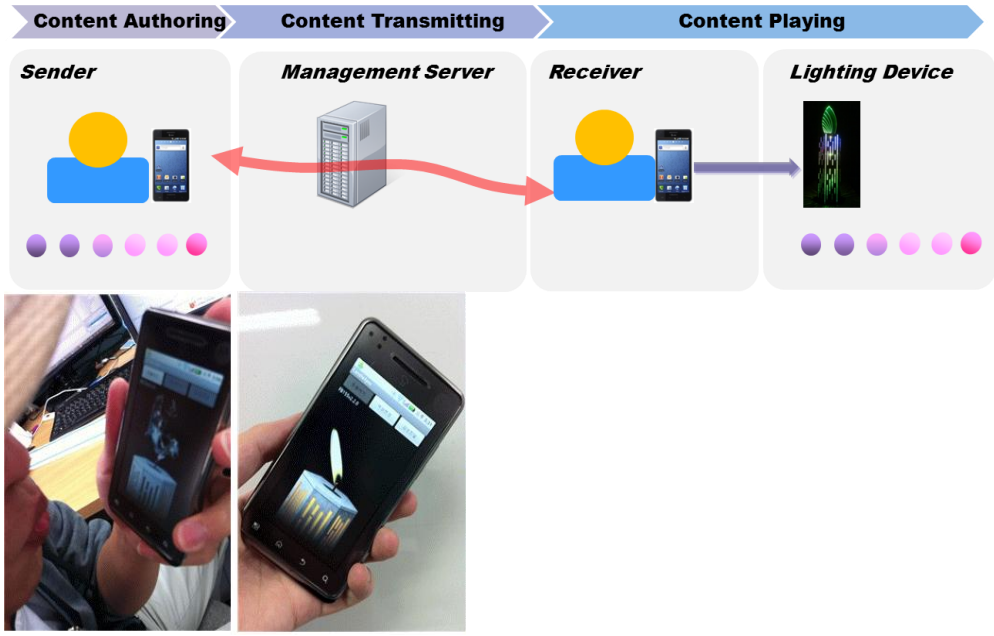
<table>
<tr><td rowspan="3">Description of each member's role in the project</td><td>Member 1
Mr. Kang</td><td>App idea
Network part programming
Content format
Sensor programming</td></tr>
<tr><td>Member 2
Mr. Park</td><td>User interface
Candle animation and candle color blending
LED HW development
Service programming</td></tr>
<tr><td>Member 3
Ms. Lee
(Not the member of CS 190)</td><td>Help for the graphical design</td></tr>
<tr><td>Estimated drawbacks and solutions</td><td colspan="2">How to make a content format
- Make the content size as small as possible.
How to send the information and awake service
- Use Bluetooth connection
- Many applications use service already and we think that there will be no problem</td></tr>
<tr><td>Schedule</td><td colspan="2">Week 4
- Overall planning
- Discuss basic stream and context of the application
Week 5
- Design of Activity and Views for Candle
Week 6
- Receive images from designer
- Insert image to the Activity
- Finish candle programming
Week 7
- Bluetooth programming and test
- Finish core module for the app
Week 8
- Enhance user interface
- Consider voice input (optional)
- Finish application
Week 9
- Test and verify
Week 10
- Prepare showcase materials</td></tr>
</table>

Attach more detailed explanation for your applications !

Weekly report

Week6	Completeness	10 %
	Main objectives and results of this week - - -	
	Contribution of team members	
	Member 1	
	Member 2	
	Member 3	
	Main results of team members	
	Member 1	
	Member 2	
	Member 3	
	Detailed summary of this week	
	Remaining issues	

Project report for CS 190

Team	Candle Sync
Members	Dong-Byeong Kang Chang-Woo Park
Application name	Candle Sync
Abstract	Candle sync is an application for LED mood lighting system. Users can send their feeling or mood through the wireless network using their smartphone. User can select a color of the candle and can send it to other user's phone. When receiving message, candle sync is awaked and display it on the screen. Also, it can be connected to real LED devices through Bluetooth.
Features	Users can send their feeling or mood through the wireless network using their smartphone. User can select a color of the candle and can send it to other user's phone. When receiving message, candle sync is awaked and display it on the screen.
Reference Apps	LED
Comparison to the reference apps	Similar to LED application, user can send the content to other's phone in the aspect of LED like display. But, in candle sync, we want to emphasize 'candle' and connect to real LED devices.
Figures, block diagrams or sketches	

Categories of App (You can have multiple checks.)	Academic aspect	Technology oriented	O
		Showcase oriented	O
		Public usage oriented	
		App store oriented	
		Etc.	
	Market aspect	Campus	
		On the go	
		Entertainment	O
		Social	O
		Game	
		Lifestyle	
		Multimedia	
		Etc.	
Self-evaluation	Level 1(Very low) – 5(Very high)	Creativity	4
		Technology	4
		Usability	3
		Market	3
		Etc.	
	Technology 0 : never used 1(Very low) – 5(Very high)	User interface	3
		Graphic design	3
		Multimedia	3
		Network	4
		Connection to social network	3
		Location	0
		Camera	0
		Augmented Reality	0
		Sensor	3
		NDK	0
		Server	0
		Connection to other devices	4
		Additional HW design	4
	Final Score	Level (0-5)	
		Technology(0-5)	
		Efforts(0-5)	
		Total (0-15)	

<table>
<tr><td rowspan="6">Contribution and description of each member's role in the project</td><td rowspan="2">Member 1
Mr. Kang</td><td>Contribution</td><td>60%</td></tr>
<tr><td colspan="2">App idea
Network part programming
Content format
Sensor programming</td></tr>
<tr><td rowspan="2">Member 2
Mr. Park</td><td>Contribution</td><td>40%</td></tr>
<tr><td colspan="2">User interface
Candle animation/candle color blending
LED HW development
Service programming</td></tr>
<tr><td>Member 3
Ms. Lee (Not the member of CS 190)</td><td colspan="2">Help for the graphical design</td></tr>
<tr><td></td><td colspan="3"></td></tr>
<tr><td>Estimated drawbacks and solutions</td><td colspan="3">How to make a content format
- Make the content size as small as possible.
How to send the information and awake service
- Use Bluetooth connection
- Many applications use service already and we think that there will be no problem</td></tr>
<tr><td>Schedule</td><td colspan="3">Week 4
- Overall planning
- Discuss basic stream and context of the application
Week 5
- Design of Activity and Views for Candle
Week 6
- Receive images from designer
- Insert image to the Activity
- Finish candle programming
Week 7
- Bluetooth programming and test
- Finish core module for the app
Week 8
- Enhance user interface
- Consider voice input (optional)
- Finish application
Week 9
- Test and verify
Week 10
- Prepare showcase materials</td></tr>
</table>

Week6	Completeness	10 %
	Main objectives and results of this week - - -	
	Contribution of team members	
	Member 1	
	Member 2	
	Member 3	
	Main results of team members	
	Member 1	
	Member 2	
	Member 3	
	Detailed summary of this week	
	Remaining issues	

Week7	Completeness	30 %
	Main objectives and results of this week - - -	
	Contribution of team members	
	Member 1	
	Member 2	
	Member 3	
	Main results of team members	
	Member 1	
	Member 2	
	Member 3	
	Detailed summary of this week	
	Remaining issues	

Week8	Completeness	50 %
	Main objectives and results of this week - - -	
	Contribution of team members	
	Member 1	
	Member 2	
	Member 3	
	Main results of team members	
	Member 1	
	Member 2	
	Member 3	
	Detailed summary of this week	
	Remaining issues	

Week9	Completeness	80 %
	Main objectives and results of this week - - -	
	Contribution of team members	
	Member 1	
	Member 2	
	Member 3	
	Main results of team members	
	Member 1	
	Member 2	
	Member 3	
	Detailed summary of this week	
	Remaining issues	

Week10	Completeness	100 %
	Main objectives and results of this week - - -	
	Contribution of team members	
	Member 1	
	Member 2	
	Member 3	
	Main results of team members	
	Member 1	
	Member 2	
	Member 3	
	Detailed summary of this week	
	Remaining issues	

Block diagram of the application

Used technology / SW module	Description (Also, open sources, SW modules, APIs used your project and so on can be explained here.)
User interface	
Graphic design	
Multimedia	
Network	
Connection to social network	
Location	
Camera	
Augmented Reality	
Sensor	
NDK	
Server	
Connection to other devices	
Additional hardware design	
Etc.	

Activity Design

Attach more detailed explanation for your applications !

CS 190 – Spring 2012

CS 190 Final Paper Guidelines

1. GENERAL

Each group is required to write a comprehensive report about the project. The report should consist of approximately **10 to 15 pages** describing the research you have done and should also include some background information, problem statement, methods and approaches, experimental results, discussion of the results and conclusions.

Your final report might be reviewed by your mentor before you turn it in. If you choose to ask your mentor to review your final report, make arrangements with your mentor to read your report and get her/his comments and modify it accordingly before your final submission.

2. FORMAT

- All reports should be readable by Adobe Acrobat or MS Word on a PC.

- It is recommended to use "Times New Roman-12" font, single-spaced with double-spaces between paragraphs and headings.

- Final report should have the following sections:

Front-page:

- Title page (project title, names & student IDs of group members, name of faculty mentors)
- Abstract
- Table of contents

Body:

- Introduction
- Background
- Main body with subsections
  - Detailed description of your design
  - The analysis of your final results
  - Problems encountered and how you solved them
  - Non-technical issues
  - Areas of improvement
  - Budget information etc.
- Conclusions
- Acknowledgments (if any)
- References

Back-page:

- Appendices (if any)

- You may use the following format for your final report:

ABSTRACT
1. INTRODUCTION (Regular numbers are preferred to Roman Numerals)
2. BACKGROUND (Regular numbers are preferred to Roman Numerals)
3. TITLE of 1st section of main body (e.g. explaining the principle, or giving some theoretical background, etc.)
   3.1 Title of 1st Sub-Heading
4. TITLE of 2nd section of main body (e.g. describing the experimental methods or the results)
   4.1 Title of 1st Sub-Heading
5. DISCUSSION AND CONCLUSIONS
6. ACKNOWLEDGMENTS

7. REFERENCES

- You are required to address at least two of the following realistic constraints:

- economic
- environmental sustainability
- manufacturability
- ethical/health/safety
- social/political

Describe how two or more these constraints relate to your project and how you will address them. (you are encouraged to discuss this with your mentor)

- All figures should be numbered and should have a clear, brief caption.

- If a heading falls at the bottom of a page, insert a page break before the heading to keep it together with the text.

- Number the pages at the bottom, center.

- It is recommended to number references consecutively (in square [ ] brackets), throughout the text.

If you refer to the same source again, use the same number. All cited sources are to be collected in a reference list at the end of your report.

It is also recommended that the references should be numbered in the order they are first used in the text. IEEE guidelines for reference format can be used:
IEEE STYLE MANUAL

- Example for print sources (periodicals, books and etc.)

[1] Author, “Name of paper,” *Abbrev. Title of Periodical,* vol. *x,* no. *x,* pp. *xxx-xxx,* Abbrev. Month, year.

[2] Author, *Title of Book, xth* ed. City of Publisher, Country if not USA: Abbrev. Name of Publisher, year, pp. xxx-xxx.

- Example for electronic (online) sources

[1] Author. (year, month day). Title. *Journal* [Type of medium]. *Volume(issue),* paging if given. Available: http://www.(URL)