courseApp

IFI course application for

Android

Final Report

By
Akbar(Shahab) Faghihi Mughaddam - shahabfm@usit.uio.no
Ummair Tahir - ummairt@ifi.uio.no
Piraba Kaugathasan - pirabak@ifi.uio.no
Shahab Bakhtiyari - shahabb@ifi.uio.no

Table of contents :

1. Introduction

Our plan in this semester project was to develop an Android mobile application which allows the users to browse easily the available courses at IFI(Institutt for informatikk) on their smart phones. We hope our work can give students enough flexibility to surf easily through all available courses, add them to their watch list, receive latest updates from the courses added, and even in later versions update their Android calender with the important dates for their chosen courses(deadlines for assignments, exams, etc).

1.1 Who are we ?

The background of our group members are different from each other, but what we share is a common desire for production and development which motivates all of us to start this challenging project. We belong to different study groups varying from Nano-Electronic and Robotics to Network and administrations and even one of us who takes free courses at IFI. What we all want here is to learn more about Android development world and learn how to use all the possibilities around it, either in later at work or in our master thesis.

1.2 Motivation

The whole idea of this work started from a small example that was shared with us during one lecture, why not a project to present course taking at IFI easier so the student do not get frustrated. Earlier experiences and later research by courseApp group members showed that courses structure at IFI are being published in a very outdated manner in a non-structured list(see Figure 1). We hope after our work no student needs to go through 190 courses at IFI to be able to find out available courses which are related to Robotics or Mobile Technologies. This dilemma more and less applies to all institutes and faculties at University of Oslo.

Figure-1

2. Process & Challenges

This project has been one of the challenging processes we have been to, as we had to do both practical work and at the same time come back to academic environment to back our assumptions and defend our choices in an academic manner. We have learnt a lot through this process and all the challenges we had. In this chapter you can read a brief story of the process of our work and the challenges we faced.

2.1 Assumptions About Target Users

Our target users are first of all students at department of informatics, at university of Oslo. This group of users are the current outdated system. This target group provides us a very good opportunity for both testing purposes and a big number of potential active users. After discussing about our target group in our developing team, we came up with following assumptions.

- The students visit course pages very often.
- They are frustrated over lack of a channel for being informed about latest updates with the courses they take.
- It's difficult to surf web pages that are not designed for mobile users (UiO course pages).
- There is a large number of students with Android smart-phones at IFI.
- Usually IT students are more eager to try new applications/gadgets.
- Informatics students have better understanding about the development process which can bring lots of useful feedback to us.

2.2 Challenges

2.2.1 Are we what we think we are ?

The first challenge was faced right after submitting the first wonder document of this project, with our ideas about how we want to polish and renew the students course taking world at IFI. The received feedback was challenging. The most critical part of it, was if we were able to prove our assumptions. We had claimed that *most of students at IFI own smart phones* and also the claimed *the majority of students are Android users*. There exist many surveys showing the rate of smart phones vs. simple phone, and also some which shows Android as the dominant operative system in the smart phone market(**SOURCE ??????**), but we decided to run our local survey to have our own academic results exactly from the target group we focus on. This resulted in our group getting involved in both qualitative and quantitative information gathering methods, to understand the target users needs and statistical data to back our assumptions. This will be discussed in details in next chapter, *Qualitative and Quantitative Research*.

2.2.2 Young Researchers

We believe through this process and with challenges we faced, we have become four young scientists that have learnt how to

- *Have an idea*
- *Think about it*
- *Research around the concept*
- *Not to be afraid to confess about being wrong*
- *Do what it takes to finish the job*

This process made us to go through scientific articles, discussing Mobile information systems and theirs development tools, and take part in seminars about the future of Application developing and cross-platforming opportunities(10). All these beside the very fruitful lectures and guest lectures we had this course(INF5261) have learnt us so much about the Mobile Computing itself and the future of it from expert's point of view. The fruit of all these are discussed in chapter 4, *Theoretical Research*.

2.2.3 Getting Access

As the primary things were set up, we could move on to the real business; courseApp for android. To make this come true, there was a need to access information about the courses. For this purpose we contacted the web section of University of Oslo where we made a query about whether we could get "read" access to the course database of university. We initially wanted to be able to at least have read access to the "last messages" from the course and get information about the lecture timings and location.

We were referred to contact with people in charge in USIT, where they finally came up with a solution that was helpful. We were not granted access to the database system, but we could use an XML-document(7) or an RSS-document(8) in order to satisfy our needs. They also mentioned that they are very interested in our project and they would like to know the results(see Appendix II for the details).

2.2.4 Technology

Android development is like a puzzle where you have to put the right tiles together in order to get working. This process at the beginning was a bit challenging, as we had to make Android SDK to talk to Eclipse and later Eclipse to have the basic fundamentals for Android development which comes with a plug-in. This process is described in chapter 5, *Practical Works*.

2.2.5 Time

The last, but not least which can be even called our biggest challenge in this work has been the factor of time as all of us are working full time and studying beside it. To agree on appointments and meet up was one of the most challenging parts of the job, as the project itself was a very interesting subject and it was fun to get the job done, but we learnt quite a lot from this process to be well structured and plan a head.

3. Qualitative and Quantitative Research

To use and interact with this big potential target group, we have decided to use a combination of both qualitative and quantitative methods to collect academic data. This would both help us to find out if our assumptions were correct, and also would let us know about the needs of our target group. As for qualitative we decided to go for interview as it is more accurate than simple observation and the discussion with interviewee can inspire us in our development process. Our favorite quantitative method was statistical survey. An easy way to collect data which we did it by making a survey application in Google Docs, and spread the participation link among students by Email. This choice of ours and the advantages and disadvantages around it are discussed in details later in this part.

It's also necessary that although we did not have any need for storing sensitive data, we have taken into account the *ethical issues* around our research. This was done by collecting all the data *anonymously* with no sensitive data being stored when it applies to our survey. This strategy has been also followed in interviews by hiding the real identity of interviewee.

3.1 Interview

Interview is always a powerful tool to both find out about the needs of potential users and at the same time get inspired by their ideas. In advance we had prepared some questions that we would like to ask the interviewee about, but at the same time we wanted it to be an open-ended interview, which means that we tried to make it more like a conversation, focusing on one specific topic (1, p.289). Under an open-ended interview it may come up new information and ideas that were not anticipated. The advantage of this method is that the interviewer(here the developer team) would be inspired by the ideas of the interviewee. A disadvantage with this method is the quantity of unstructured data that are generated. This can be very time-consuming to analyze (1, p.289).
For this phase of our project we have interviewed three students at departments of informatics. Following points is the summary of their needs and ideas for such application(Detailed information about the interviews can be found under Appendix I). We should mention in advance that some of these needs or better said wishes can not be implemented in an application without necessary infrastructures and new services from both departments of informatics and university of Oslo(Video streaming of lectures, more standards about course pages, etc). Therefore the findings are divided in to two groups, called *realistic* and *wishes*.

Realistic :

- Easy surfing through courses
- Categorizing and tagging courses not only by institutes but also technology(robotics, image processing, design, etc)
- Update Android calendar with the deadlines and important dates about the desired courses
- Warnings about both important dates and new updates on screen(Notification)
- Latest updates on the course pages
- Support for several platforms and not just Android

Wishes :

- Ability to join courses on the app(Join button)
- Ability to download course presentation
- Possible video streaming of lectures in future

3.2 Statistical Survey

After submitting the first wonder document of this project, we received some challenging feedback. The most critical feedback was to prove our claims about *most of students at IFI own smart phones* and also the claim saying that *the majority of them are Android*

users. There exist many surveys showing the rate of smart phones vs. simple phone, and also some which shows Android as the dominant operative system in the smart phone market, but we have decided to run our own local survey to have our own academic results exactly from our own target group.
Statistical survey is a obtrusive method for collecting systematic information from a group of individuals. Surveys are important sources of information for all kinds of research fields, varying from marketing research, psychology up to health services.

> *An obtrusive procedures require a person's cooperation. Examples of obtrusive procedures are interviews, achievement tests, and questioners(11, P. 183).*

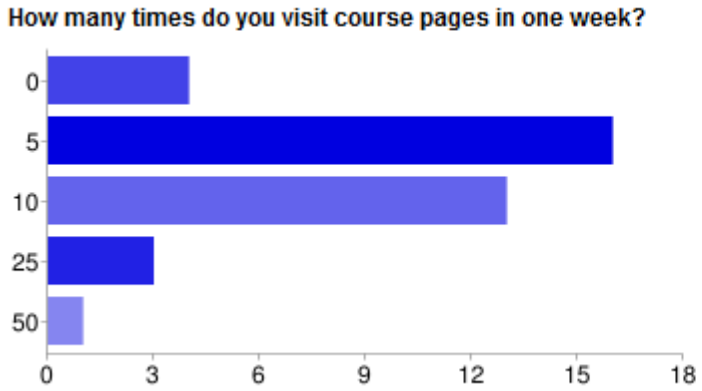
The survey method in this project is *online surveys*, as it is a method which is know to be easy to implement and administrate, low coast, little time consuming and *very fast* as the researchers have access to the feedback right after participators have clicked on send button. We are also aware of the disadvantages with this method like the issue of honesty in answering and the role of manipulation of result by Internet robots(We tried to avoid simple HTML forms and use Google tools to prevent this one as much as we could).
As mentioned earlier this online survey was developed using Google Docs and the participation link was later distributed by Email to all of the students we knew, and again redistributed by them to other IFI students. We believe the result from this academic research can back our assumptions.

3.3 Results

Figure - 1

As per today, 37 individuals have responded to the survey we made. The graph, as Figure-1 elaborates gained its peak the very first day where we had 12 respondents and later flattened out. We can also see that every now and then, somebody does answer to our survey queries as they either haven't had time earlier or they have forgotten the mail-link. This survey is as earlier told not a replication of all the students as our target group right now is limited

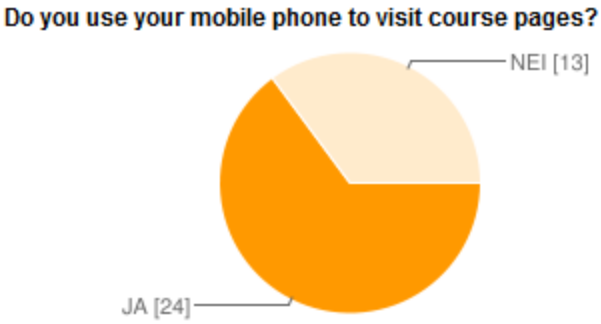
to the students at the institute of informatics.



0	**4**	11%
5	**16**	43%
10	**13**	35%
25	**3**	8%
50	**1**	3%

Figure-2

What we can be sure of from Figure-2 is that the course pages are frequently used. Majority of respondents visit the course pages of their respective subject more than 12 times a week.



JA	**24**	65%
NEI	**13**	35%

Figure-3

The figure above shows that a good 65 percent of those who have participated in the survey use their mobile devices to visit the course-pages, which is a sign for us.

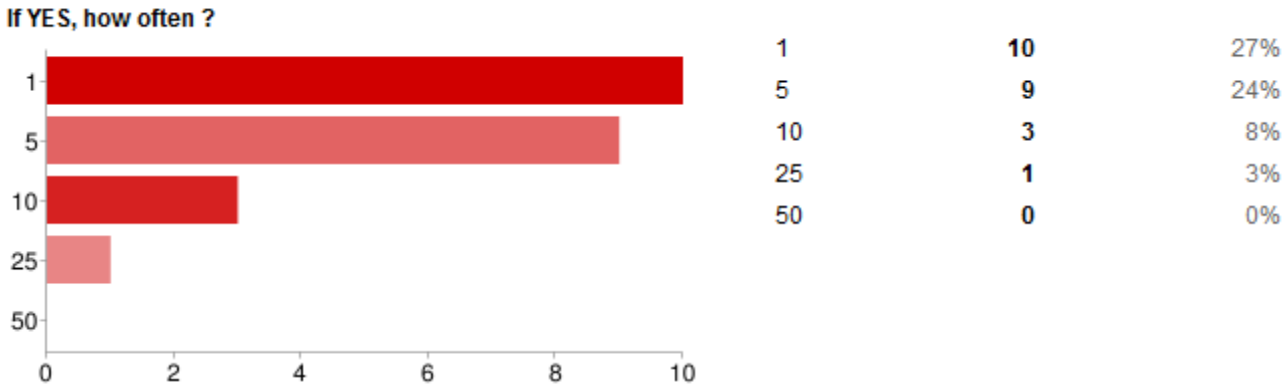


Figure-4

Also in the fourth question in our survey, we got a positive response. A good majority used their mobile device more than 8 times a week to keep updated. The question about how satisfied the respondents were with the presentation of the site on their devices. Having said that, as all members of our group are using smart-phones we could ourselves go to any course page and conclude that the the page became a little too small for our phones, even for a Samsung galaxy S with 4 inch screen, zoom has to be used.

Figure-5

In the era of 2000, smart devices have made their march into the consumer market and as shown by the answers of the respondents an overwhelming number of participants have smart-phones.

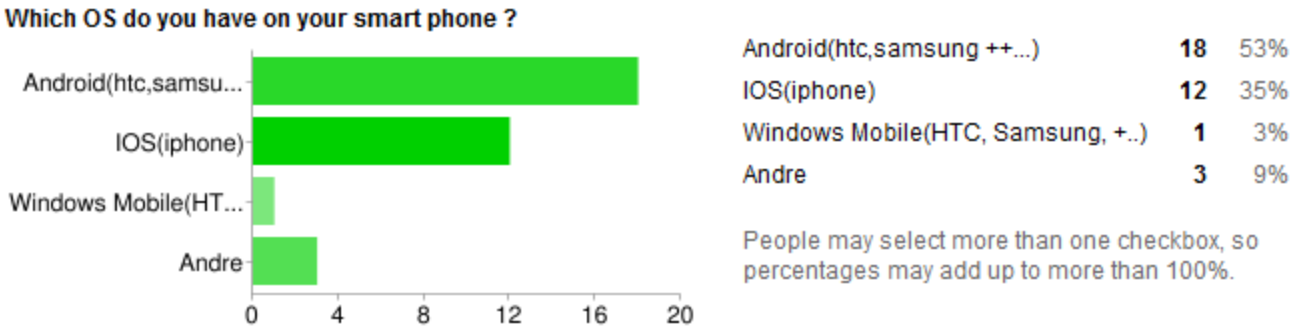


Figure- 6

This question and the result after thirty seven individual answers supports our assumption about the majority of IFI students having Android OS on their smart-phones. Most of the users have devices with android operating system, which also is a good replication of how the situation is in the market now. We know that the number of android users are growing(12) which made the choice of platform easier for us.

4. Theoretical Research

4.1 Choice of Programming Language

There are several programming languages available for developing on mobile devices varying from HTML and XML to different versions of Java and C family. The factors that are important in choosing the right programming language for developing an application for mobile devices can be divided in following categories :

- *Which platform ?*
- *Efficient use of current resources*
- *Cross-Platform development*
- *Speed and performance / Complexity*

As mentioned earlier in previous chapter, our survey showed that Android is the most popular platform among IFI students. This and other practical reasons resulted in Android, a Linux based operative system, to be our chosen platform. This narrows down our choices to a handful number of programming languages which are HTML/X-HTML/ HTML5(mixed with other web technologies), Java running on Dalvik and native C. In following sections we compare our choices regarding *Cross-Platform development* and *Complexity, speed and performance* of the different programming languages that can serve in developing an application for Android. The last section is trying to underline that every project should consider to reuse all the available resources both in terms of

financial manners, developers experties, available development platforms and present raw materials(e.g. on Internet). This is discussed in the last section of this chapter where we tried to argument our choice.

Table 1. Required Skill Sets for Nine Mobile OSes

Mobile OS Type	Skill Set Require
Apple iOS	C, Objective C
Google Android	Java (Harmony flavored, Dalvik VM)
RIM BlackBerry	Java (J2ME flavored)
Symbian	C, C++, Python, HTML/CSS/JS
Windows Mobile	.NET
Window 7 Phone	.NET
HP Palm webOS	HTML/CSS/JS
MeeGo	C, C++, HTML/CSS/JS
Samsung bada	C++

Figure 7(13)

4.2 Cross-Platform

With the recent revolution in mobile technology and having multiple mobile platforms like iOS, Android, Blackberry, WP7 and others in the front of the bleeding edge technology, the cross platform question is once again a relevant topic.

How do you build software that can run on Blackberry, iPhone, Android, and even Windows Phone 7?

The endless technological war between Apple (iPhone/iPad), Adobe (Flash), Google (Android), and Microsoft (Silverlight, Windows, Windows Phone) has more losers than winners; Apple products will probably never run Flash, Google will never run Microsoft's Silverlight, and Microsoft will never promote Java or Flash.

The war has a lot of casualties; many of them are the developers, trying to find a way to support Android, iPhone, Blackberry, and soon, Windows Phone 7.

But without noticing, from the ruining of this crazy war, raised a new force. Not exactly "new" but more of a "renewed" force. *Completely neutral*. It is HTML 5. It's the classic good old web technologies, including Html, Javascript, those basic technologies responsible for billions of web sites and web applications all over the Internet, those technologies have quietly emerged from building simple static pages to creating basic apps, they evolved form enabling simple on-line apps into creating advanced on-line/ offline apps for any platform out there. Including mobile.

4.3 Complexity, Speed and Performance

As discussed earlier in choice of programming language section we do have HTML family, Java(running on Dalvik) and native C in our choice domain. This section tries to compare these three regarding speed and performance. We first go through how HTML5 and other technologies can help us resolving the speed and performance problem in mobile development, and later move on to Java and native C. In later part we use results from scientific findings to show the difference between native C and Java running on Dalvik in speed and performance in Android platforms.

4.3.1 HTML Family

When it comes to the choice of speed and simplicity, HTML family and other web based technologies, like Javascript, PHP and etc, sounds to be the right one. The reason lies on the simplicity, experience and the server side compiling of the of the code which spares us lots of CPU cycles on our phones, something that we already lack. This is an obvious fact that day after day, our daily life depends more on mobile devices and our mobile systems have to process more information. Specially when it comes to brand new HTML5, this web technology gives developer great advantages in graphics layout, user interaction and resolving many of today's issues about streaming of multimedia to mobile devices.

There are many positive aspects that HTML5 gives as the language for web design and web development.

Advantages

- **Offline Support** – There are some features that come with HTML5, which allow you to store more information Offline (more than cookies / cache) which is a huge advantage for developers of mobile applications. This will allow useful applications to continue running even when no Internet connection.
- **Web and Video** – These features make it easier than ever to add graphics (not really too big of a problem before) and video broadcasting (large front) to a web page and make inter- compatible mobile browser.
- **GeoLocation API** – This is not really part of HTML5 but since HTML5 is mainly related to the use of mobile devices, geolocation is consistent with this discussion quite well. Developers can now more easily integrate location-specific user information based on an application and do things like this "augmented reality cool stuff people have spoken. That's how it is possible to point your iPhone or Android device to a street and tell you where the nearest store coffee is (ever wondered why your iPhone requires you to use your "current location "?).
- **Advanced Forms** – Now your mobile browser can handle some basic things

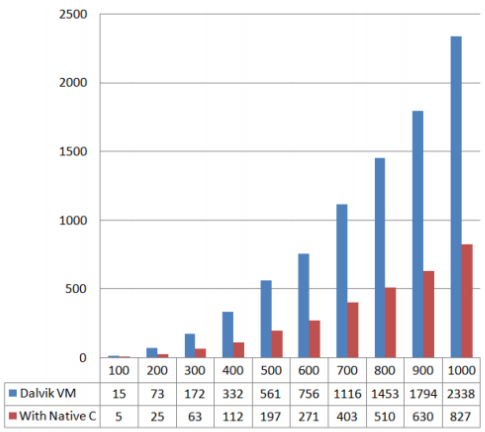
necessary to ensure that the information you enter into an online form is correct (form validator) instead of having to run additional scripts for do. This will really speed up loading times and increase things like experience utilisateur.

Disadvantages

- **Lack of Support** - HTML5 is not yet fully supported by Internet and besides that, there are few other deal breakers "that will hold up HTML5 to be fully integrated into all platforms immediately . The biggest problem is the acceptance of micro-formats on multiple browsers.
- **No Backward Compatibility** - HTML5 uses a lot of new labels that add new semantic richness to a Web documents, but does not fully utilize micro-formats (class attributes, etc.) which makes the backward compatibility of HTML5 nonexistent. Being a web designer, it's a bad chose.

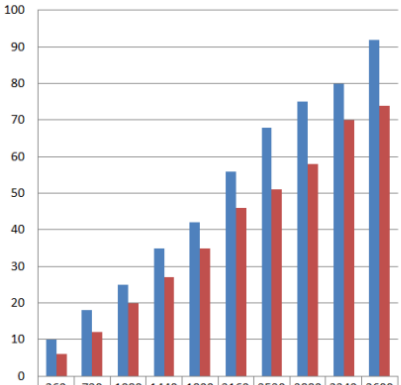
4.3.2 Java(Dalvik VM) and Native C

Java with its special virtual machine for Android(Dalvik) when is compared to native C gives very interesting results. Sangchul Lee and Jae W. Jeon in an article presented in International Conference on Control, Automation and Systems 2010 (14) compared these two development methods on one Android Virtual Device emulator. The experiments in this article have compared these two programming methods in terms of speed, in different situation such as Integer and float-point calculation, memory access and heap memory allocation. You can see the results in Figure 8. The blue columns stand for the time that Dalvik Virtual Machine(Java) takes to do the operation while the red columns stand for the same amount of time that it takes for Native C to compute the same operation.



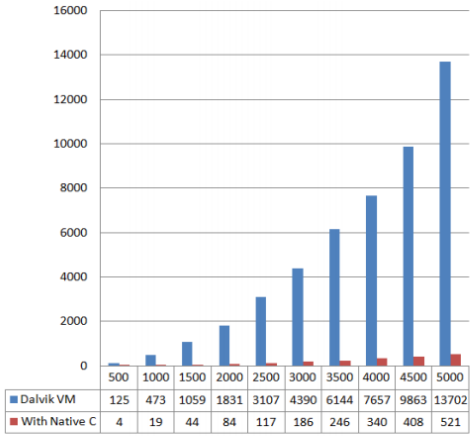
	100	200	300	400	500	600	700	800	900	1000
Dalvik VM	15	73	172	332	561	756	1116	1453	1794	2338
With Native C	5	25	63	112	197	271	403	510	630	827

Fig.4 Results of the integer calculation experiment.



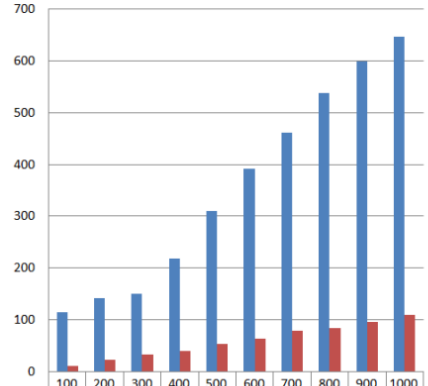
	360	720	1080	1440	1800	2160	2520	2880	3240	3600
Dalvik VM	10	18	25	35	42	56	68	75	80	92
With Native C	6	12	20	27	35	46	51	58	70	74

Fig.5 Results of the floating-point calculation



	500	1000	1500	2000	2500	3000	3500	4000	4500	5000
Dalvik VM	125	473	1059	1831	3107	4390	6144	7657	9863	13702
With Native C	4	19	44	84	117	186	246	340	408	521

Fig.6 Results of the memory access algorithm experiment.



	100	200	300	400	500	600	700	800	900	1000
Dalvik VM	114	142	150	218	311	392	461	538	599	647
With Native C	11	22	33	40	54	64	78	83	96	109

Fig.7 Results of the heap memory allocation algorithm experiment.

Figure 8(14)

The high complexity of Native C programming is noted several times in this article and the authors underline that the speed should be considerably faster so it is worth the effort to code a project with Native C.

> *"Android application developers need to consider the advantages and disadvantages of using native code. Using native code does not always enhance application performance, but it always increases application complexity(14)."*

Moreover the analysis of these raw data form the experiments clearly shows that no matter which scenario, Native C is a faster approach to compute operations in Android virtual emulator with AMD Athlon II X2 245 2.9GHz CPU. The design of different scenarios, which mentioned earlier, are very critical here as the factor of speed varies a lot in each case. Sorted according to speed difference from low to high in different cases

Task Name	X factor faster in Native C	If it is worth the complexity
Float-Point Calculation	Less than 2 times	No
Integer Calculation	Around 3 times	No
Heap memory allocation	Around 6 times	Might be
Memory access	More than 31 times	Definitely yes

Table of speed comparison showing how faster Native C is in front of Java running on Dalvik VM(14)

As the table up here shows, the speed factor is different from each scenario to the other one and that's why Sangchul Lee and Jae W. Jeon clearly bring this sentence in the conclusion of their work :

> *"We recommend that Android application developers use the native C library when the application frequently requires memory access and that needs complex calculations under correct usage(14)."*

4.4 Why Java running on Dalvik VM ?

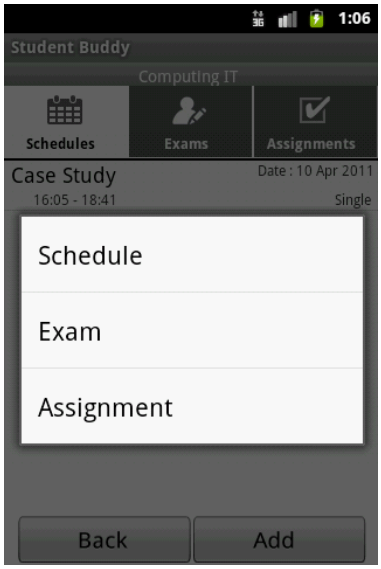
The choice for Java(Dalvik VM) was taken due to many reasons, listed below.

- Google official development tool is Java running on Dalvik, native C is recommended in special cases
- All of the group members had expertise in Java
- C is much more complex to program and we do not have much memory access in our code
- Everyone in our group had Android smart-phones
- Eclipse combined with Android SDK are free development tools
- HTML 5 is not quite fully developed yet and not all web browsers are supporting it yet

4.5 Previous Works

Our search at Android Market resulted in some programs with almost the same functionality as our project. With these application can all student organize all theirs activities. You can track all the deadlines and plan your assignments using this application.Also adds holidays and the instructors of the classes, import, backup and password protection.

Student Buddy

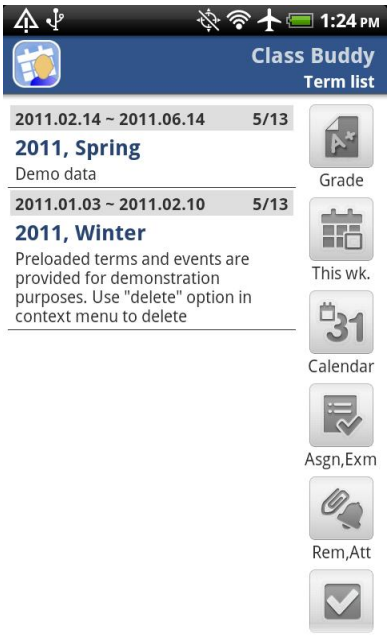


Student Buddy(15)

Features:

- Today view in the home screen automatically displays summary of current classes and tasks after basic schedule was input.
- A very detailed schedule planner.
- Interactive multifunctional calendar including expanded month and week views.
- Differentiate all the courses using colour labels.
- Distinguish all the class types using predefined set of around 30 icons.

Class Buddy Pro



Class Buddy Pro(16)

Features:

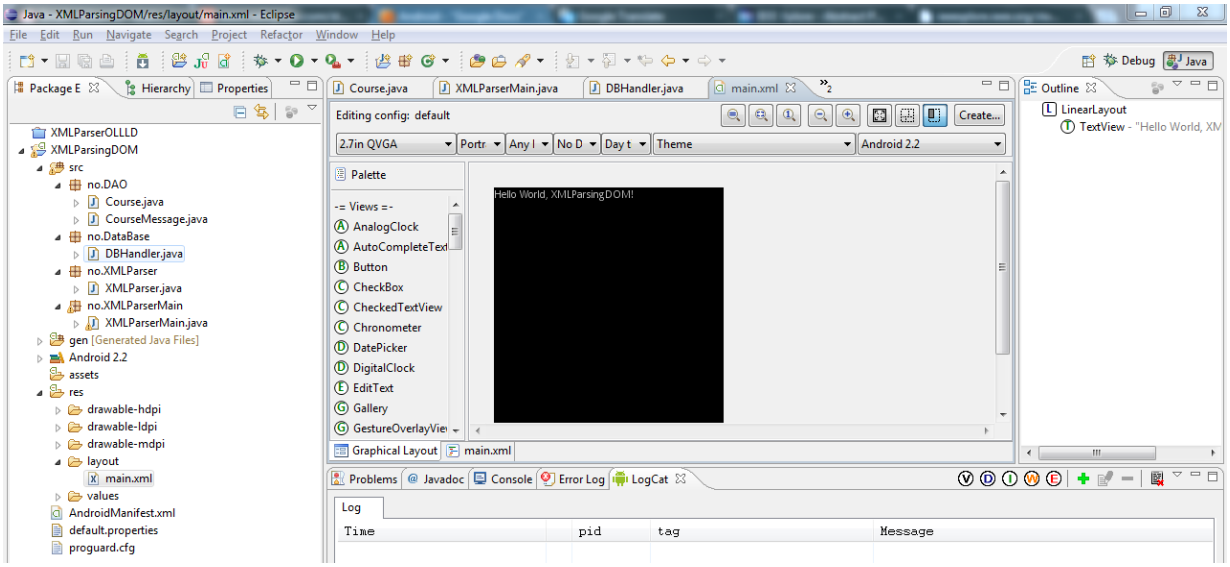
- Organize your: term,semester, courses, events such as class meetings, assignment, exam, quiz, etc...
- Scores
- Grade and GPA
- Expenses
- Import and Export(CVS,google doc,google calender), auto-Sync to Google calendar
- Customizable Home screen Widget
- Simple task manager
- Backup and restore
- Reminder
- Password protection

Our CourseApp is more customized to be used by UiO students. Since our program can find info from UiO's web pages directly which benefits directly UiO students.

5. Practical work

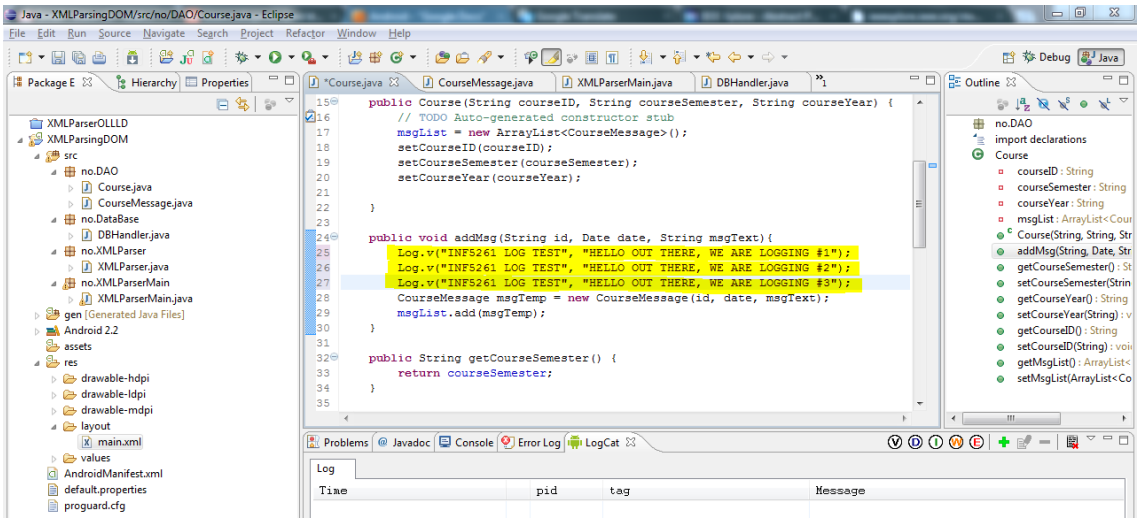
5.1 Tools

In order to develop any application on Android operative system, one needs to install Android SDK(2). The Android SDK in addition to Eclipse IDE(3) is the most recommended choice to start development phase. We used the latest version of Eclipse IDE for java developers(4). In addition to these two, there is also need for isntalling an extra plugin in Eclipse to connect the IDE to the Android SDK. This plugin is called ADT plugin for Eclipse(5) developed by Android SDK developing team.



Eclipse IDE, Android GUI builder

As none of us had any experience with Android programming, and like most programmers we sat down after graving the development kit, and wrote a “Hello World” application. It was a good start and we did run it by the Android emulator. This was quite a slow process from pushing the run button to getting merely the short text “Hello World” printed on the screen. During development of this prototype we encountered several different ways to do things and by the way found our own favorite development tools. As mentioned earlier, the slowness of emulator was a major problem. This resulted in that we were debugging our code on our phones(using hardware devices, or debug using bridge). In this method you connect your phone to the computer where you are coding in Eclipse. You should enable the USB debugging feature first(Setting-Application-Development-USB Debugging). After connecting, the Android SDK recognized the phone and the code can be tested on the phone. When enjoying the fast speed of testing your code on the phone you miss the a very great tool called Logger on the Emulator, but there is always a way around the problems. LogCat is a tool that can be used to monitor every single thing that happens on the phone and you can use it as a equivalent to *System.out.println()* in Java SE for debugging your system. This little tool was one of our biggest discoveries since you can easily print out information and double check if everything is going as planned.



LogCat(17) - One of the most useful tools you can hav while developing for Android

It was also worth mentioning that in case of *sudden crash* which can happen quite often, as every source usage should be declared in advance, it is a good idea to have a good logging system to go through(permissions issue will be discussed in next section).

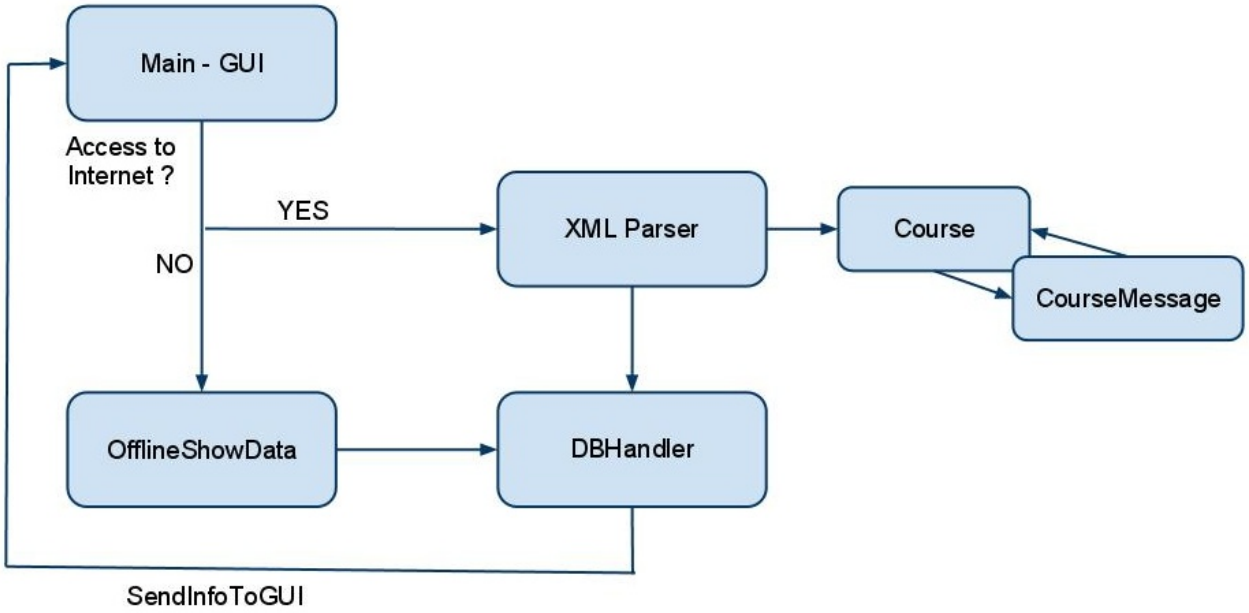
5.2 Technology

The next challenge was to find out the best way to read this pile of information in the best possible way. We came across a number of different technologies to parse XML documents and acquire information. Some of the common XML parsers are SAX, XML PULL, Feed Parser and DOM parser(9). After trying out the SAX parser and DOM parser. As the choice of the XML-parser depends on which kind of application one wants to develop, we decided to go for the DOM-Parser because it was favourable for using when one had to read from smaller documents(9). After a few failed attempts we finally succeeded in parsing the XML document for the course code INF3100(7,8) and got it to work. It is worth mentioning that one needs to make some declaration in the file *manifest.xml* which is found in the root of the project folder. In this file you can define which resources(components) of the device, the application would have access to. This issue is called permissions and is very vital while developing. In case of forgetting to mention permissions in advance, the program will crash on your device with no understandable error code. It is here that using the emulator can be a huge help, as there is a a good logging system present, which can help you to find out the reason for crash.

5.3 Prototype, Ready

As per now, we have tested all the technologies needed in this project and right now we have a functional XML parser application that can read all the latest updates from

any course at IFI. This prototype enjoys stable functionality, while it has very little user friendly graphic user interface. This application was shown to several students at IFI and the response even for the prototype was more than expected, as they asked us if they could get it on their own phones, or even we were asked by iPhone users, if there is a corresponding application with similar features for iPhone.



CourseApp UML Diagram

Right now we have finished the coding of XML Parser, Data Access Objects, and Database Handler and we should connect these parts to the Graphic User Interface. We hope to have our final design with GUI before presentation day.

6. Further Work

Our goal as mentioned is to provide a better course management and viewing application and we will try our best to deliver at least a functionality which enables the students to hold track of their courses in an easy way. Having said that, we have a limited time schedule to complete this application and we will probably not be able to implement all the functionalities that we have planned to. For development in the future by both the departments of informatics and the other students who may want to add more modules to our work, we can advice the following features which are either our own ideas or are extracted from the research which we have done in this field on students at IFI.

6.1 Wider Research

There should be conducted a survey with more questions. This will give us a much wider perspective than done for this project. USIT can perhaps be requested to distribute the survey to all the students from different faculties of the university. Our application can be linked to for downloading so that the participants can try it and give their verdict. This will give a much broader output and make it easier to get an idea of things which are missing from our applications.

6.2 Further Development

Our group has primarily been focusing on the functionality and simplicity of this application. If time allows, we will try to make the GUI more colourful and elegant, but there will always be room for betterment in this segment.

Earlier in *Theoretical Research chapter* we have discussed various types of languages and their compatibilities with other platforms and also optimization with respect to speed and performance. Maybe when other developers start looking at this application HTML5 is finished and therefore would be more suitable choice for developing cross-platform applications. There can also emerge other technologies in the future which can be useful in enhancing the performance of this application.

6.3 A Note to USIT

In certain areas, a cooperation from USIT will be needed as it is they who can change the settings on the server side of the course sites.

6.3.1 Structuring the courses

Student feel the need of a better categorizing and structuring of the course lists. Categorizing courses and tagging them by technologies can be advised. This should be done by authorities at IFI or USIT.

6.3.2 Student web access(Join button)

As per now, there is no access for the students to log on to the student web through an android application. We are planning to contact the web-section of UiO at USIT, in order to be able to provide this functionality in later versions.

6.3.3 Video streaming of courses on your device

There have been rumours that in the future there will be podcast of the lectures and maybe video streaming. These all can be played on such device.

6.3.4 Improve the definition of XML rules

The department of informatics should consider defining more restrict XML tags for all

important dates and headings. An example we have hit in is that the date for examination in a course is given in XML format while the date and timings for submitting assignments are not.

7. Conclusion

As per now, we have tested all the technologies needed in this project and right now we have a functional XML parser application which lacks a good graphic user interface. In the near future, we will come up with a interactive, user-friendly and easy in navigation. For this we have to do some study and research i.e for choosing colours, layout, font-colors etc. Further development of a user friendly GUI, testing it on test users and finally publish the application on the Android market, where every student at IFI can download it, are our final steps.

8. References

1. Preece Jenny, Rogers Yvonne, Sharp Helen, (2007): Interaction Design – beyond human-computer interaction, John Wiley & Sons Ltd
2. Android SDK : http://developer.android.com/sdk/installing.html
3. Eclipse Foundation : http://www.eclipse.org/org/
4. Eclipse IDE for java developers : http://www.eclipse.org/downloads/packages/eclipse-ide-java-developers/heliossr2
5. ADT plugin for Eclipse : http://developer.android.com/sdk/eclipse-adt.html
6. Using Hardware Devices, Android Developers website : http://developer.android.com/guide/developing/device.html
7. XML sample, UiO website : http://www.uio.no/studier/emner/matnat/ifi/INF3100/v11/beskjeder.xml
8. RSS sample, UiO website : http://www.uio.no/studier/emner/matnat/ifi/INF3100/v11/beskjeder.xml
9. IBM tutorial for XML parsing on Android : http://www.ibm.com/developerworks/opensource/library/x-android/
10. Navet student group seminar at IFI about *HTML5 and future of App development* hold by Den Norske Dataforening
11. Herman J. Ader, David J. Hand, Gideon J. Mellenbergh, (2008) : Advising on Research Methods: A Consultant's Companionby
12. Nielsen Wire, Apple Leads Smartphone Race, while Android Attracts Most Recent Customers http://blog.nielsen.com/nielsenwire/online_mobile/apple-leads-smartphone-race-while-android-attracts-most-recent-customers/

13. Mobile Application Development: Web vs. Native
14. Sangchul Lee; Jae Wook Jeon; , "Evaluating performance of Android platform using native C for embedded systems," *Control Automation and Systems (ICCAS), 2010 International Conference on* , vol., no., pp.1160-1163, 27-30 Oct. 2010 - URL: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5669738&isnumber=5669634
15. Student Buddy : URL : https://market.android.com/details?id=com.vimukti.studentbuddy.ui&feature=search_result
16. Class Buddy Pro - URL : https://market.android.com/details?id=com.acj0.classbuddypro&feature=search_result
17. Android Developers, LogCat
URL : http://developer.android.com/guide/developing/tools/logcat.html

9. Appendix

9.1 Appendix I

The interviewees were three students at departments of informatics at university of Oslo.

Student 1 :

Nickname : Kian

Age : 23

Sex : Male

Interview date : 10.03.2011

Duration of interview : 10 minutes

Place of interview : Physics building

Method : Note taking and no recording as we believe recording would make interviewees uncomfortable

Notes from interview :

The interviewee owns a Samsng Galaxy S smart-phone which runs Android 2.2.

He is not familiar with concept of programming for mobile devices but he believes it is a very interesting concept.

Kian believes our app is very handy and he would even liked to install our prototype, which could get the latest updates from course pages, on his own smart-phone and become one of our test users.

The interviewee would like easier access to information about the courses as he believes it is not easy to find out things about courses at IFI right now.

The interviewee also mentioned integration with Android calendar for the deadlines and important dates about the desired courses. He mentioned that he would be even happier with warnings on the screen about deadlines and exam dates.

The subject of the weeks lecture is something he would like to see on the warnings.

Downloading course presentations on his device is another feature that he could use a lot.

Video Streaming of lectures on his device was another possible feature for him.

Student 2 :
Nickname : Thomas
Age : 27
Sex : Male
Interview date : 12.03.2011
Duration of interview : 8 minutes
Place of interview : Ole J. Dahl house
Method : Note taking and no recording as we believe recording would make interviewees uncomfortable
Notes from interview :
The interviewee owns a iPhone 4.0 smart-phone which runs IOS.
He is not familiar with concept of programming for mobile devices but he believes it is a very interesting concept.
Thomas believes our app is very useful and he wishes a similar application for iPhone.
The interviewee would like to join courses through his cell phone, or his future iPad.
The necessity of integration with calender on the device and the concept of notification/ warning were emphasised by the interviewee several times. He specially would like to have notification on his screen when he receives new updates from courses.

Student 3 :
Nickname : Maria
Age : 24
Sex : Female
Interview date : 14.03.2011
Duration of interview : 5 minutes
Method : Note taking and no recording as we believe recording would make interviewees uncomfortable
Notes from interview :
The interviewee owns a HTC Desire Z smart-phone which runs Android 2.2.
She is familiar with concept of programming for mobile devices and she believes it is a very interesting that someone finally is doing this for IFI students.
Maria believes the integration with calender on the device and the concept of notification/ warning would make her everyday at school much easier.
She complained a lot about the lack of structure in IFI courses which frustrate her every time she wants to choose courses for the coming semester.
She believes also that it is a good idea to know who else is taking the course(if any of her friends is taking that course)

9.2 Appendix II

Email text stored as a picture from web section at USIT to us which granted access to read XML and RSS databases.

Heisann.

For å hente beskjeder fra emner, kan du se XMLen fra emnene slik:
http://www.uio.no/studier/emner/matnat/ifi/INF3100/v11/beskjeder.xml?vrtx=source
eller som RSS
http://www.uio.no/studier/emner/matnat/ifi/INF3100/v11/beskjeder.xml?vrtx=feed

Tid og sted for undervisning finnes idag i 2 løsninger.
Den ene er per idag kun tilgjengelig som HTML og XML
eks.:
http://www.uio.no/studier/emner/matnat/ifi/INF3100/v11/tid-og-sted.xml?vrtx=source

mens den andre har et noe annet format og kalender funksjonalitet:
http://www.uio.no/studier/emner/uv/pfi/PED2100/v11/tid-og-sted.html

Det kommer til å skje en del endringer med UiOs Studieinformasjon
fremover, og vi kan ikke garantere at disse tjenestene vil fortsette slik
videre; og heller ikke garantere for stabiliteten.

Fint om du kan holde oss oppdatert med hvordan denne appen blir, slik at
vet hvordan den leser mot nettstedet vårt (gjerne med å svare på denne
eposten).

Lykke til med utviklingen!