

Apéndice A

Master Thesis Project in Computer Science “Interactive Kiosk for the Rolex Learning Center”, CRAFT research lab (EPFL), Lausanne (Suiza), October 2010



ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Interactive Kiosk for the Rolex Learning Center

Master Thesis Project in Computer Science

CRAFT research lab

Lausanne, October 2010

Javier Bello Ruiz
Candidate

Dr. Frédéric Kaplan
Thesis Director

Interactive Kiosk for the Rolex Learning Center

©Copyright by Javier Bello Ruiz 2010
All rights reserved

Lausanne, October 2010

Keywords

Human-Computer Interaction, Interactive Furniture, Touch Screen, Graphical User Interface, Radio-Frequency Identification, Video Messages.

Abstract

The vision of the EPFL President Patrick Aebischer is

to build a university where traditional boundaries between faculties are replaced by a spirit of collaboration; a campus designed in such a way that mathematicians and engineers may meet spontaneously with neuroscientists and microtechnicians to envision new technologies that improve everyday life; a campus that is open and welcoming to the public. (Lausanne, 2010)

The recently inaugurated Rolex Learning Center is the materialization of that concept, in this building open to the public, the contact between students, teachers and visitors is creating a new paradigm of interaction and collaboration in which people of several different disciplines share the same space, discovering together the possibilities of working in common for future development.

CRAFT research lab shares this vision and is very much involved in the training policy of the EPFL and in the spirit of the computer-supported collaboration. This computer-supported collaboration learning (CSCL) [10] is taking shape in numerous devices like the Reflect table [8], the Lantern [9] and many others that are going to be soon part of the normal life of the university.

In this context, this Master Thesis Project is aimed to develop a new prototype of interactive kiosk for the Rolex Learning Center that should accomplish two major goals, being open for everyone that visit the building and providing useful information in terms of an interactive platform that allows people to contact each other in this framework.

We use touch gestures as a natural approach to the interaction with the device that includes a vertical Projected Capacitive Touch (PCT) screen, an audiovisual system composed by two speakers and a camera with an integrated microphone, and finally a Radio-Frequency IDentification (RFID) reader capable of recognizing student cards. Although this interactive furniture could offer a large set of applications in the future, this project is aimed on building a service of video messages and a few simple information applications keeping in mind the terms of interaction and collaboration. The system relies on a Rich Internet Application (RIA), device controllers and other necessary software.

A simple user study was conducted at the end of the project, evaluating the usability and the relevance of the kiosk prototype with positive results, complemented by the logs and statistics stored into the system and the feedback provided by the users.

Acknowledgments

I would like to thank all CRAFT members, specially to Frédéric Kaplan for his supervision and advice, Pierre Dillenbourg for allowing me to be part of his team, Olivier Guédat for building the nice totem of the kiosk and all of them for their help.

To all Erasmus students that I met this unforgettable year, all my friends and family.

Dedication

To my mother. She was gone too soon, she did so much for us.

Table of contents

1	Introduction	1
1.1	Context and goals of the work	1
1.2	Working environment	2
1.3	Outline of the contents	3
2	Analysis	5
2.1	Study of the problem	5
2.2	Initial study of the audience and use cases	6
2.3	System functionalities and features	10
2.4	System architecture layers	11
2.4.1	Interface	11
2.4.2	Logic	12
2.4.3	Persistence	14
2.5	Software: decisions and components, used technologies	15
2.6	Hardware: decisions and components, used technologies	17
2.7	Discarded features	18

3	Design	19
3.1	Selected prototype and first evaluation	19
3.2	System architecture layers	21
3.2.1	Interface	22
3.2.2	Logic	22
3.2.3	Persistence	30
3.3	Software: decisions and components, used technologies	31
3.4	Hardware: decisions and components, used technologies	34
4	Deployment, system test and evaluation	38
4.1	Final deployed design	38
4.2	Detected problems	50
4.3	Maintenance and management	51
4.4	Use study: experimental results, qualitative analysis	52
5	Conclusions	60
5.1	Personal review	60
5.2	More relevant aspects	61
5.3	Improvements, further applications	61
5.4	Qualitative analysis of logs and feedback	62
A	User's manual and development guide	65
B	Use study questionnaire and consent form	74
	Glossary	79
	Bibliography	81

List of Figures

2.1	State diagram of the system	9
2.2	Interface prototype	13
2.3	Interface hierarchy	14
2.4	Entity-Relationship diagram of the database	16
3.1	First prototype of the video message board application in blue and red colors	23
3.2	Modified prototype of the video message board application in white colors, trying to follow the design of the Rolex Learning Center	24
3.3	Final prototype of the video message board application with more videos per page and better quality colors following the design of the Rolex Learning Center	25
3.4	First prototype of the virtual touch keyboard	26
3.5	Extended prototype of the virtual touch keyboard	27
3.6	Process for identifying the user through the RFID reader and the Web Services	28
3.7	LDAP directory structure	29
3.8	Final chosen webcam with integrated microphone	34
3.9	RFID UART reader	35
3.10	RFID UART interface	36
3.11	Connection between the RFID UART reader and the RFID interface for the PC	37

4.1	Final hardware arrangement in the kiosk's totem back	39
4.2	Final built and working prototype of the Interactive Kiosk for the Rolex Learning Center	40
4.3	Totem back detail	41
4.4	Main screen	42
4.5	Apps screen	43
4.6	About, terms and privacy policy	44
4.7	Restaurant menus app	45
4.8	Metro timetable app	46
4.9	Feedback app	47
4.10	Today's question app	48
4.11	Video message board app	49
4.12	Number of users per day in the use study: probable impact in the logs of the video message board	54
4.13	Most useful applications according to the use study question- naires	54
4.14	Least useful applications according to the use study question- naires	55
4.15	Most liked applications according to the use study questionnaires	55
4.16	Most disliked applications according to the use study ques- tionnaires	56
4.17	Opinion on the usability of the system	56
4.18	Opinion on the interface and device design	57
4.19	Opinion on the audiovisual quality in the video message board	57
4.20	Problems observed during the free exploration of the kiosk . .	58
4.21	Problems reported by the users during the use study of the kiosk	58

4.22	Evaluation of the usability in the task steps of recording a video message on the video message board during the use study of the kiosk	59
5.1	Logs of the video message board application: played, recorded and contacted number of videos in the period from 15/07/2010 to 15/08/2010	63
A.1	Flash Builder 4 Flex compiler configuration	67
A.2	RFID reader command format	68
A.3	RFID reader command list	69

Chapter 1

Introduction

In this first chapter of the thesis we are going to give a general view of the subject faced, putting it into the development context and giving the summary of the contents that we are going to explore lately in the document.

1.1 Context and goals of the work

This Master Project was carried out at CRAFT (Centre de Recherche et d'Appui pour la Formation et ses Technologies) research laboratory of the EPFL (École Polytechnique Fédérale de Lausanne, Switzerland) under direct supervision of Dr. Frédéric Kaplan and advice of Professor Pierre Dillenbourg, under the student exchange Erasmus programme of Javier Bello Ruiz from the Centro Politécnico Superior (University of Zaragoza, Spain).

This project is aimed to build a hardware prototype system of an interactive kiosk, to program an initial set of useful applications for people in the Rolex Learning Center and integrate them within a reliable software framework and interface to be developed.

This interactive furniture was made in a way that it supplies a solution for actual needs of people, helping them to perform some every day task like searching for accommodation, looking at the metro timetables and deciding about the cafeteria menus of the day. In a future, as the framework of applications is very wide can also be a source of needs, i.e. developing an EPFL interactive TV where the only place that you can participate is at Rolex Learning Center, making the building even more attractive to people.

The goals of this project are the following:

- Having a reliable system because of the fact that is exposed to many people and has to show robustness against errors.
- The system should rely on the concepts of usability and accessibility.
- Having useful tools that people would like to use in this environment.
- Being fully portable to a server in order to have more than one kiosk.
- Having an user-friendly interface with an attractive design that allows people to interact with the kiosk and among them.

1.2 Working environment

As the device is placed into the university campus, it needs multilingual support due to the diversity of languages spoken in Switzerland (Swiss German, French, Italian and Romansh) and the huge number of foreign students, teachers and researchers.

Students should be able to easily record video messages in front of the kiosk and to browse other student's messages. The system will permit to easily obtain contact information about a given message by simply placing the student's card over the RFID reader. Other small applications should be developed in order to provide interesting services for the students.

Different software components should be built for this project: a Flex-based interface, a C++ controller for the RFID reader, a MySQL database and a Hypertext Preprocessor (PHP) architecture for linking the different layers of persistence, logic and interface.

In order to accomplish all these objectives we conducted an initial survey among several students in order to know which applications are interesting for this kind of device analyzing their proposals. Then we decided about the different hardware components needed for using them into the prototype, testing them and choosing the most suitable ones. We also decided the orientation of the screen based on making the interaction user-friendly, deciding which orientation fits with the initial set of applications considering pros and cons of having maps or virtual keyboards for example.

We decided between different programming languages, web/media servers, operating system environment, supporting software and we developed the software system capable of providing these applications and integrate the

suitable functionalities. Regarding the physical process of building the furniture support and electronics needs, this work was carried out by Olivier Guédât, one of the CRAFT engineers.

1.3 Outline of the contents

After this introduction we are going to find four more chapters organized as follows:

- *Chapter 2: Analysis*

Based on the results and data collected from observations in the initial survey among the students and previous knowledge on HCI (Human-Computer Interaction) and GUI (graphical user interfaces), this chapter focus on the analysis carried out to determine an appropriate solution for a prototype of the kiosk. We analyzed the different technologies to be used, the hardware needed, the applications that we can provide in the kiosk and can be more attractive because they don't have them in other platforms like portable devices. This analysis is structured in a three-layer system architecture based on the paradigm of MVC(Model View Controller [12]) in which we defined the layers of interface, logic and persistence.

- *Chapter 3: Design*

Based on the analysis and following the same structure, we design a solution that can supply the previously defined goals and objectives. In this chapter then we describe the different prototypes and their design with their features, problems and improvements.

- *Chapter 4: Deployment, system test and evaluation*

The final prototype is deployed in a real usage scenario that involves its target users. Evaluation processes will be performed throughout this period to collect and analyze the impact of the work. This will integrate users' feedback and suggestions, logs of the system and finally an use study in which ten people participated giving us very useful information about the system. This use study relies on the thinking aloud protocol [11].

- *Chapter 5: Conclusions*

One of the most important chapters, showing the conclusions reached after the analysis of the results. I included a personal review of the work, the most relevant aspects observed during the deployment and

the improvements or further applications that we can make to the system based on the analysis of the strong and weak points of the kiosk.

After the main body of this thesis we are going to find the following appendix:

- *Appendix*
 - A User's manual and development guide
 - B Use study questionnaire and consent form
- *Glossary*
- *Bibliography*

Chapter 2

Analysis

In this chapter we are going to face the study of the problem of creating a new interactive kiosk. Analyzing the different possibilities in terms of software and hardware environment, several decisions were taken on the features of the system.

2.1 Study of the problem

For facing the goal of having an operative kiosk at the end of the project, the work plan was based on a common procedure involving three phases: analysis, design and deployment. At the end of the third phase the conclusions about the final device are given by the tests of the system and the evaluation. All this processes follow a user-centered approach.

Our work will first start from an initial study of the applications suggested by a certain number of the students at the EPFL community and our own ideas about suitable apps that should fit in this kind of prototype. The results of this observation phase will be analyzed to figure out what would be needed to support those works and interactions. Throughout the work, user's perspective will be integrated to help foster and shape prototype interface and functionalities. Experiments and evaluation methods will be also extensively utilized.

At the beginning we had to decide between different things, the main idea was to build an interactive information kiosk to be placed in the Rolex Learning Center but there were a lot of possibilities. We had to decide on the interaction: making it able to recognize gestures, only work with a touch approach or merging both approaches; decide on the language and

language environment: Flash (Flex), C++, OpenGL, GTK, Java/Processing, Silverlight, etc; decide which operating system is the most suitable; and finally which type of services we had to implement in order to offer an useful system: web server, administration application, apps browser, apps player, CMS blocks, templates, customizable Really Simple Syndication (RSS) feed, customizable Keyhole Markup Language (KML) maps, video player, kiosk avatar, SDK, etc. All of that without forgetting that the app should be accessible in a cloud based way for being able to have several kiosks.

2.2 Initial study of the audience and use cases

One of the most important things at the very first moment on the project, was being aware of the different needs of people using this kind of device at the university. For that reason, we conducted an initial study of the audience in which we asked several students about the applications that they consider useful in an interactive kiosk at Rolex Learning Center. The result of this study was the following list of applications:

- Plan of the Rolex Learning Center, interactively controlled with different touches or gestures in case of gesture recognition.
- One application for finding or offering accommodation. As finding accommodation in Lausanne is quite difficult, they consider an advantage to have an interactive panel of offers accessible in the building.
- As the main library is in the building, it would be useful to consult media sources like books, magazines, comics, CDs, DVDs or any other electronic information provided by the library.
- Creating personal events in the kiosk making them visible for other students.
- Buy and sell posts application.
- Video posting for searching a foreign language tandem between students.
- Digital blackboard for brainstorming.
- Videoconferencing, like using Skype, with available teachers or with important researchers invited to events.
- Video e-mails: an easy way to record a restricted length video to be sent with the EPFL mail account.

- Lausanne, EPFL and Rolex Learning Center events.
- Cafeteria menus of the day.
- Bike renting at the EPFL.
- Timetable for the metros arriving to or leaving from the university, next arrivals and departures.
- Special offers of flights.
- Weather forecast.
- Internet access by gesture browsing or using the touch screen.
- Information about the sports in the UNIL sport center.
- TV channels.
- Mobile phone refilling.
- CAMIPRO card refilling (Student card with an electronic budget for the cafeterias).
- Consulting the database of publications, virtual library.
- Splited screen with a keyboard and the relevant information.
- Games for one or two players.
- News panel and the possibility of rating this news.
- Buy stuff of the EPFL.
- Chat with the other kiosks.
- Posts with videos, photos, text, etc. from Internet.
- Share student's calendar (like Google calendar).
- Personal account, with personal arranged interface.
- Multiple languages.
- Share to Twitter, Facebook, etc.
- Moodle.epfl.ch
- EPFL Wikipedia built by the students.
- Forum.
- Groups.

- Connected to printers.
- Music application.

We made a state-diagram in order to figure out how should be the system that can run this type of applications as you can see on Figure 2.1.

And then we made the following use-cases to determine in which one we should focus as main application and what we should need to implement that:

Use case: Search for a place in the RLC plan.

Actor: User.

Goal: Look at the plan searching for a place or information and finding it.

Description: A user browses between the different apps, selects Rolex Learning Center (RLC) plan and uses a search bar with a virtual keyboard or moves the plan with touch gestures until finding the place. After that, he closes the app.

Needs: app browser, app player, touch screen.

Use case: Browse between different apps.

Actor: User.

Goal: Exploring the different apps.

Description: When a user approaches the device, it should recognize him with person detection or using the touch screen. The user browses between the apps with the appropriate technique.

Needs: app browser, app player, touch screen, camera, person recognition, touch recognition.

Use case: Configuring the RSS aggregator.

Actor: User.

Goal: Customize the personal aggregator.

Description: After browsing between the different apps and selecting the RSS aggregator, the user has to identify himself in order to customize his personal aggregator by means of face recognition or login or student card. This aggregator offers the user the possibility of browsing the web and adding RSS feeds or disposing them with touches or gestures.

Needs: app browser, app player, web browser, touch screen, camera, database (storage about the user options), RFID.

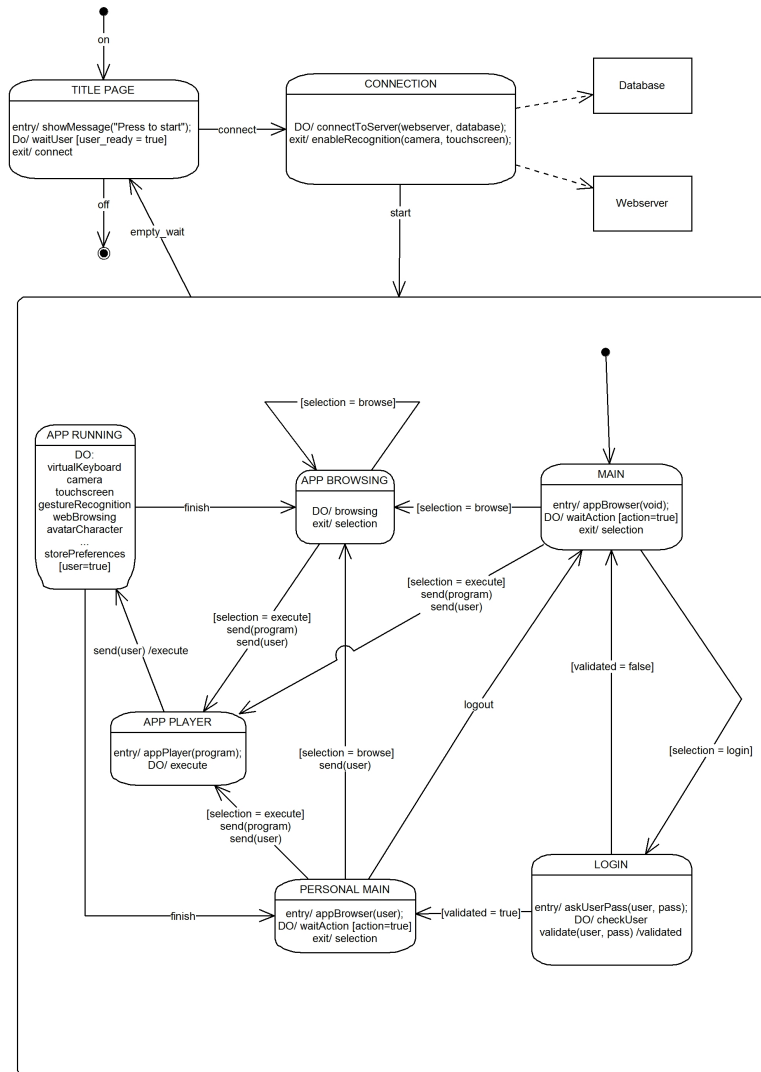


Figure 2.1: State diagram of the system

Use case: Video player/recorder.

Actor: User.

Goal: Play previous recorded videos, record his own video.

Description: After selecting the video player app, the user can play a video from a panel with miniatures of previous recorded videos, he can use a search bar with key words or browse looking at the titles. He can record his own video and add a title.

Needs: app browser, app player, touch screen, speakers, microphone, camera.

Use case: Interaction with the kiosk avatar.

Actor: User.

Goal: App that uses the avatar.

Description: When the user wants to use a gesture recognition approach in order to interact with the kiosk, the system can provide an avatar reflected in the screen representing the user for making him able to see which buttons he is touching. When a user decides to explore the kiosk avatar functionalities, a reflected avatar appears on the screen and the interface changes according to the new interaction, that allows the interaction without touching the screen.

Needs: app browser, app player, gesture recognition, camera.

Use case: Adding tags to the RLC plan.

Actor: User.

Goal: Add tags to the places on the plan and make comments, add info or name a new place, etc.

Description: Editor in the RLC plan for adding tags to the places by writing them with the virtual keyboard, and storing them. The user can choose between see the map with or without tags. This tags should be approved in some way by an administrator of the system.

Needs: app browser, app player, admin, touchscreen, database, KML, Spatial Features Oracle.

We analyzed carefully these use-cases and the diagram in order to decide about the system as follows in the next sections.

2.3 System functionalities and features

After looking at the initial study, the use cases and the state diagram of the system it was decided to develop a system with a touch screen, an audiovisual system, a RFID card reader for identifying the users and a main application of a video message board.

This main application of video messages is able to record, contact with the user and store the videos of the holders of a CAMIPRO card, that includes students, PhD students, teachers and other personal of the EPFL. We picked this option among the others because it was a way to merge several of them. In a video message board you can play and record videos, you can leave messages recorded for the others in order to search for accommodation, language tandem etc. because each video belongs to a category that you can choose. It was a way to offer several possibilities considered by the students in the initial survey in only one application.

This application was developed in a Flash-based environment because we searched the goal of have an independent system, with the possibility of being executed in several machines through a server. Flex and in this case Flash Builder offers the possibility of developing a RIA executed in the web browser that makes the app be executed in a cloud based way. In terms of the prototype, we are going to execute the application in the same machine as the server but in future it can be easily executed in other machines.

This RIA development tool works with Flex SDK: ActionScript 3.0 and Macromedia eXtensible Markup Language (MXML). As we have to keep the information of the video messages, a database is necessary and also a logic layer to connect the interface with it. In further sections we are going to explain how Flex can interact with a MySQL database or the identification system via PHP and any other features that we discovered necessary during the design.

2.4 System architecture layers

In this section of the analysis we are going to get into the architecture of the system describing each layer in which we worked during the design and posterior deployment of the system.

2.4.1 Interface

The first thing in which we worked was the interface because we already had a PCT screen to work with it. That made us able to define the first approaches to the design of the interface that we improved during the design phase. In order to start thinking about the interface, we first decided the orientation of the screen due to the possibility of having it vertically or horizontally placed.

Regarding at the main application selected, in which a list of videos had to be shown and an app browser had to be designed, the most suitable

position of the screen was vertical, in a similar way of a smartphone, as this type of interfaces are a very widespread type of interaction and people is used to use them then it seemed to be a good approach to make people comfortable with the kiosk.

We took into consideration that the components in the screen should be clear enough to make the user quickly understand the process of performing a task without a long learning time and as less interaction as possible measured in clicks or different sections.

This took us to the first GUI prototype (Figure 2.2) designed with the help of the Adobe Flex Component Explorer [4] and the Adobe Flex Style Explorer [5]. We also thought about the hierarchy of the sections including other small information applications that we included lately in the design as we can see on Figure 2.3.

We worked a lot in the design of the different sections as you can see on next chapter.

2.4.2 Logic

The purpose of this layer is to communicate between the interface and the physical devices or the database. As Flex is not able to connect to physical ports of the PC we had to develop a way to communicate the interface with the RFID reader. A controller was developed in C++ in order to make the device read the appropriate tags of the CAMIPRO cards and send this tag to a server side program wrote in PHP responsible of the communication with the Flex application. PHP also provides the necessities packages to work with the MySQL database server for the storage of the videos and users.

In other side, for executing the application into the web browser it was necessary to consider which web server to install and if it was necessary any other technologies to provide the video streaming, in our case we discovered the need of a media server for the flash applications in order to provide this service. In the software section of this analysis we will see the different possibilities that we considered.

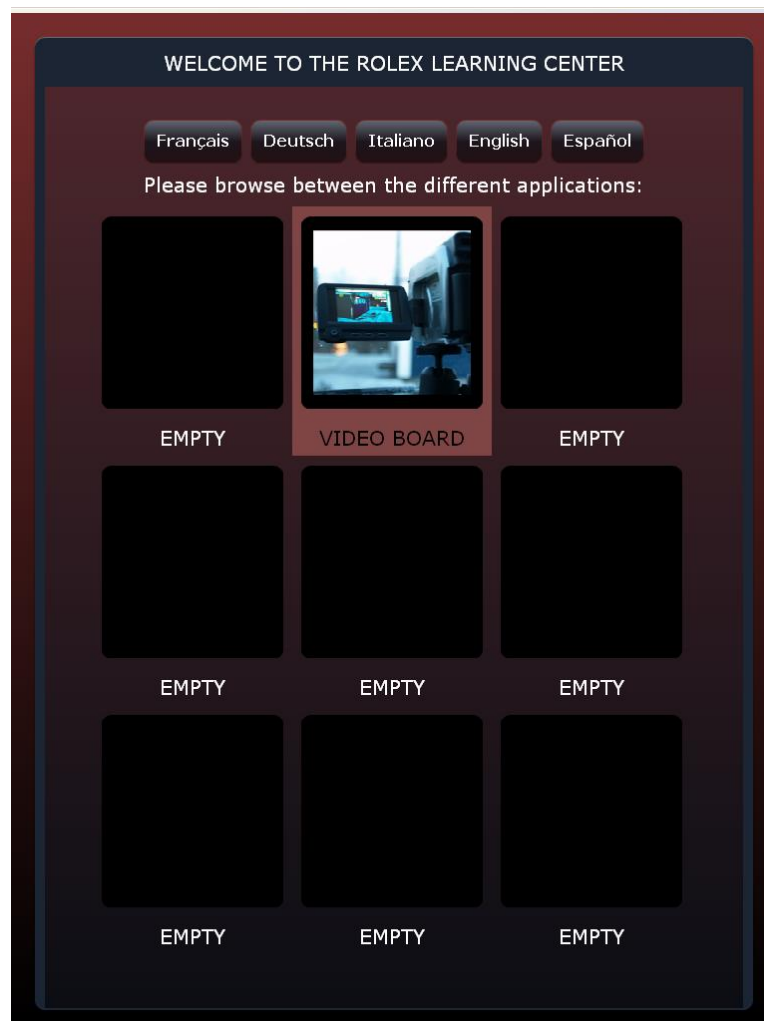


Figure 2.2: Interface prototype

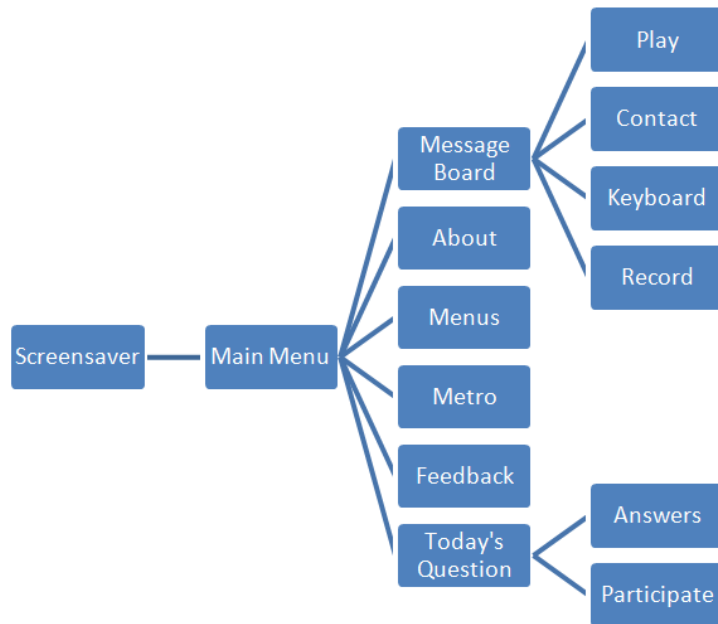


Figure 2.3: Interface hierarchy

2.4.3 Persistence

After taking the decision of having as main application in the system the video message board, several questions in terms of storage about users and video messages information came into our minds. We have already developed the use case regarding this application, and we just made an storyboard describing the interface, so the final step is knowing which information about the video messages and the users we have to store. We identified the following entities that might have our persistence layer:

- *Message* The message is the principal entity due that we have to store it for being able to display it again when required, we also have to define which are the attributes of this entity. First of all, we have to give it a unique ID for identifying the message among the others. This entity has to contain the text of the message on some way. The same thing should happen with the title displayed and with the time of recording as it is draw in one of the storyboards. Following the storyboard we have another attribute that is the number of answers. And as the video file has to be attached to this message we have to know the name of this file.

- *Person* We decided that for recording a video message and for contacting with the author, if we are interested in the message, we have to identify ourselves with the CAMIPRO card. We are going to need the data of the author and probably of the person who wants to contact him in order to send an email. So, which is the information we would like to store? First of all the unique ID that can be the SCIPER number associated with the CAMIPRO card and also we need the email in order to contact them and probably we would like to know the complete name to address the email properly in case of answer.
- *Category* The category of the messages can be an attribute or can be an entity, we think that we have a better representation as an entity because despite of the initial fixed number of categories, these categories can evolve in time to more categories or merging between them or simply despairing, it is something to consider at the end of the project. For the moment we store them with a unique identifier and with the number of messages belonging to this category.

So once defined the different entities we can provide an Entity-Relationship diagram [13] of the future database (Figure 2.4).

2.5 Software: decisions and components, used technologies

Taking into consideration the system we wanted to develop, it was very important to take good decisions about the software to use and knowing if the environments to use were capable of all the things we needed.

The main decision was the principal language to use, we decided then to use Flex SDK with ActionScript 3.0 and MXML, so the best tool in order to work with this SDK was Adobe Flash Builder 4 Premium Edition that is a professional-grade development tool designed to help software developers rapidly build cross-platform, rich Internet applications (RIA) and content using the open source Flex framework to access Web Services, REST APIs, and other data sources. We extensively used the help provided by Adobe [6].

For the database we had to make the decision between ORACLE or MySQL: due to the widespread system of MySQL and his common use in web environments was the reason of the choice. A common tool for joining the databases and Flash applications is PHP, there exists as many tutorials as Flash so it was really easy to get into this language. In the install of PHP it was necessary to include the support packages of MySQL, PEAR

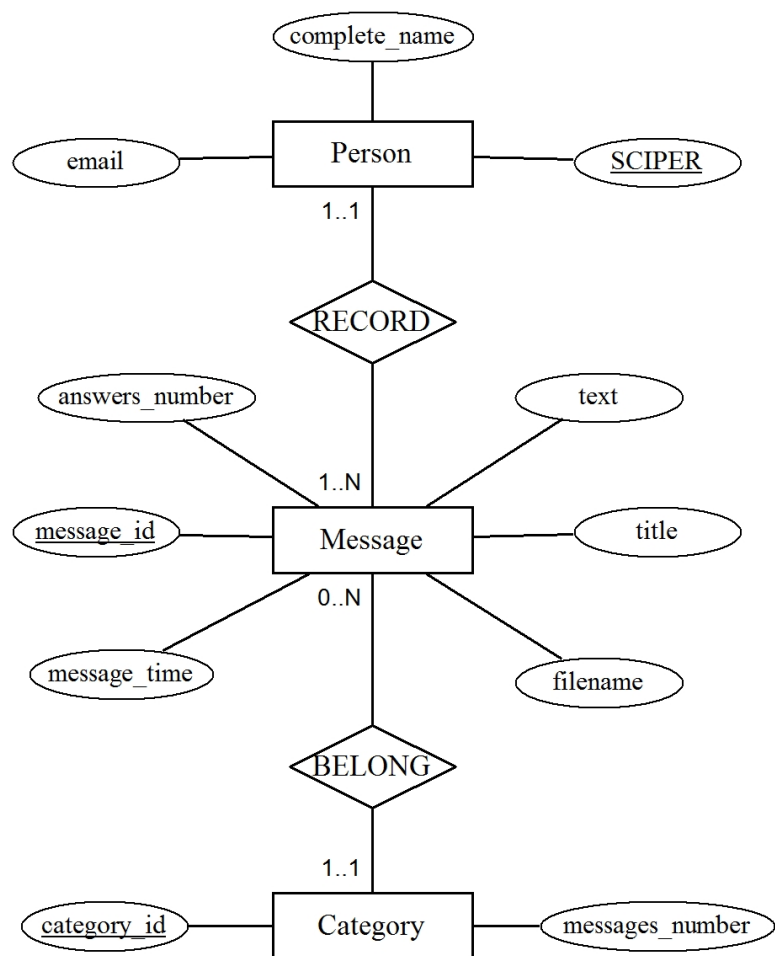


Figure 2.4: Entity-Relationship diagram of the database

for SMTP mail, Lightweight Directory Access Protocol (LDAP) and Simple Object Access Protocol (SOAP).

The PHP Extension and Application Repository, or PEAR, is a repository of PHP software code. Stig S. Bakken founded the PEAR project in 1999 to promote the re-use of code that performs common functions. The project seeks to provide a structured library of code, maintain a system for distributing code and for managing code packages, and promote a standard coding style. Though community-driven, the PEAR project has a PEAR Group which serves as the governing body and takes care of administrative tasks. Each PEAR code package comprises an independent project under the PEAR umbrella. It has its own development team, versioning-control and documentation.

As we figured out the system like a web application, we had to think about a web server, the most common used ones are APACHE and IIS, but our choice was Apache for having an easy way to configure it and a quick installation. It exists the possibility of download servers with integrated services like PHP or MySQL but we chose to install these services separately and then configure them together.

2.6 Hardware: decisions and components, used technologies

The first hardware decisions taken in the analysis phase were to find the appropriate graphics card for rotation purposes and touch screen compatibility. When we made the decision about the graphics card or a motherboard with integrated graphics card, we looked at the recommendation of the touch screen manual:

Silicon Motion Graphics

Starting with UPDD version 3.8.42 integral support for the Silicon Motion Graphics chipset is provided. The UPDD software automatically recognizes the presence of this chipset and enables the interface so no user action is required other than to select the rotation mode using the standard SM interface. This interface is understood to work with all current SM chipsets including the SM712, SM722, SM731, SM500 and Lynx (including Lynx EM) product.

ATI and NVIDIA video chipsets

Latest ATI and NVIDIA video chips, with rotation capabilities are supported.

Mobile Intel Rotate enabled Chipset Family

Starting with UPDD version 4.0.2 integral support for the Intel rotate enabled chipset is provided. The UPDD software automatically recognizes the presence of this chipset and enables the interface so no user action is required other than to select the rotation mode using the standard Intel interface. Note that the UPDD rotate interface requires the Intel rotate software to be present, which may not always be the case. It is possible that the recognition method will also recognize other Intel chipsets, however we have only verified that this method of recognition identifies the 915GM/GMS, 910GML, 855GME and Extreme Graphics 2 (Ver. 6.14.10.4497) chipset but it may recognize others.

In the end we chose a motherboard with the graphics card integrated because of the size and power, it was portable and reliable because it was used in other similar devices in the laboratory before:

Motherboard ZOTAC ION ATX-A-E mini-ITX.

Chipset: NVIDIA MCP7A-ION Series. NVIDIA nView multi-display (up to 2 monitors).

Size: Mini-ITX form factor of 6.69 inch x 6.69 inch (171 mm x 171 mm).

Microprocessor: Intel ATOM N330 (Dual-Core) CPU.

2.7 Discarded features

In the end, the gesture recognition was not included in the system development because the main application of video messages was too complex for making the interaction gesture-controlled. However as we are going to see in the conclusions chapter, as the small applications are being used a lot, it can be a solution for them, and also the face recognition in order to rearrange things on the screen.

Chapter 3

Design

Based on the previous analysis we are going to design a solution that can supply the previously defined goals and objectives. In this chapter we are going to describe the different prototypes and their design with their features, problems and improvements.

3.1 Selected prototype and first evaluation

The selected prototype of the kiosk is composed by a touch screen, a RFID reader, a webcam with integrated microphone, two speakers, a mini PC and the furniture totem. The main application of the kiosk is a video message board, but there are other four more applications: the metro timetable app, the restaurant menu application, the today's question app and the feedback application.

- *Video message board application* Application in which you can record one video message identifying yourself with the CAMIPRO card and play or contact other video messages arranged by different categories: Sell/Buy, People, Offers, Culture, Help Needed, Accommodation and Various.

The videos are arranged into the screen as a list of miniatures with the contact information and text of the message, you can either play them or press the contact button in order to automatically send an email to the author by identifying yourself with the CAMIPRO card, you can also browse between the different categories or search a video with the search bar.

In the record section you can record and store your own video message by filling the different inputs and identifying yourself with the CAMIPRO card.

We used Flash Media Server (FMS) in order to play and record the videos, the database to store them and the PEAR package to send the mails via PHP and SMTP. This app also store statistics and logs of the system that we can analyze after and they are used as information charts in the screen saver.

- *Metro timetable application* Application in which you can check next arrivals/departures of the metro to/from the EPFL. For getting the information in real time the company of the metro provides a QR-code in each stop that you can use with your smartphone in order to reach a webpage with the info. Parsing this webpage it was the approach of having the information into this application.
- *Restaurant menu application* Application in which you can check all the menus of the different cafeterias and restaurants of the EPFL. This application is built like a RSS feeder with the corresponding RSS feeds of each cafeteria.
- *Today's question application* Application in which you can either answer the today's question or submit a new one knowing the statistics of the question of the day.
- *Feedback application* Application in which you can provide feedback about the kiosk.

This selected prototype was the result of the previous analysis and first evaluation of the proposed ideas and initial design. In this first evaluation and validation of the kiosk by Frédéric Kaplan and Pierre Dillenbourg their first impressions were good but we needed to move forward and design a few more things and improvements:

- Support for several languages: French, German, Italian, Spanish and English.
- Add more videos per page in the video message board application.
- Add a screen saver to make people being interested in the kiosk when idle.
- Change theme colors to fit more on the design of the Rolex Learning Center.

- Add small applications to the system.
- New interface arrangements.

Between the different applications that we considered to implement in this revision to extend the main video app were:

- Access to the website moodle.epfl.ch
- Access to the website is-academia.epfl.ch
- Access to the website my.epfl.ch
- SBB train schedule
- Metro timetables
- Feedback section with people comments

During a second revision the following ones were selected as we showed before:

- Feedback section with people comments.
- Question of the day.
- Restaurant menus RSS.
- Metro timetables.

The decisions regarding the totem building were taken at the same time that this evaluation. We agreed on building a totem with a protector glass for the screen, with the position of the camera and speakers on the top, and the RFID reader on the right side. We were thinking about the high of the device for a long time, because it should be not too high for recording people of different highness and also not to low for not making the screen uncomfortable to use. We will get deep into these considerations at the conclusions chapter because issues of ergonomics related with these decisions appeared during the evaluation of the system.

3.2 System architecture layers

In this section of the design we are going to get into the architecture of the system describing each layer in which we worked during the design and posterior deployment of the system.

3.2.1 Interface

The design of the interface was changed after the first evaluation considering that the colors should follow the style of the building, making them brighter, smoother and clearer. For this change we defined another Cascading Style Sheets (CSS) regarding colors, shapes and other features to be applied to the design of buttons, panels and other different components taking part on the interface. In the following figures (Figure 3.1, Figure 3.2, Figure 3.3) it can be seen the evolution of the mentioned theme for the interface towards the final design.

We extended the first virtual keyboard prototype (Figure 3.4) based on the Updated Flex Qwerty Component (www.kerkness.ca) into a more suitable one for messages (Figure 3.5).

For developing the multilingual support we took benefit of one characteristic of the Adobe Flash Builder platform, you can use the different locales of the platform for automatically translate the predefined texts into the selected language. Flash Builder also provides the support for including extra locale folders for the texts defined by the developer. In different folders you can create a file named `kiosk.properties` in which you define in one side the keywords to use by the resource manager into the program and in the other side the texts translated into the different languages, the compiler make the rest compiling each different file into the matching locale.

3.2.2 Logic

In the first approach of the controller for the RFID reader, we considered the possibility of transfer the information read by the device with a socket wrote on C# connected to a client on an ActionScript 3.0 module. This solution was unsuccessful due to the security sandbox, in which Flash apps relies on, was blocking the transmission. As ActionScript 3.0 has no direct support for accessing the PC ports, other possibility was buying a RFID reader called Phidgets who has support libraries for ActionScript 3.0, but the frequency of the reader was not suitable for reading the protocol of the student cards. In the end we manage to communicate a C++ controller based on the source provided by the manufacturer RFIDGeek via PHP server-side with the ActionScript 3.0 module [7].

After managing to get data from the reader, this C++ controller can identify the ID tag belonging to the CAMIPRO card. Once you have this information is necessary to access a SOAP Web Service in order to get the SCIPER number associated with this card and person on the EPFL but was

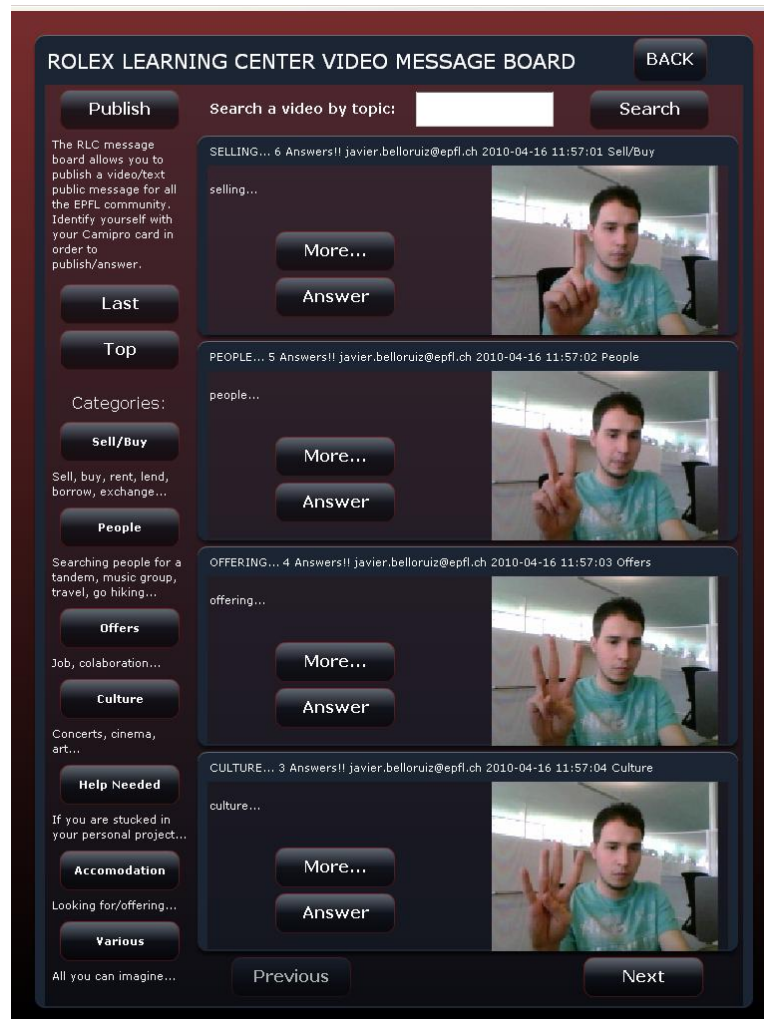


Figure 3.1: First prototype of the video message board application in blue and red colors

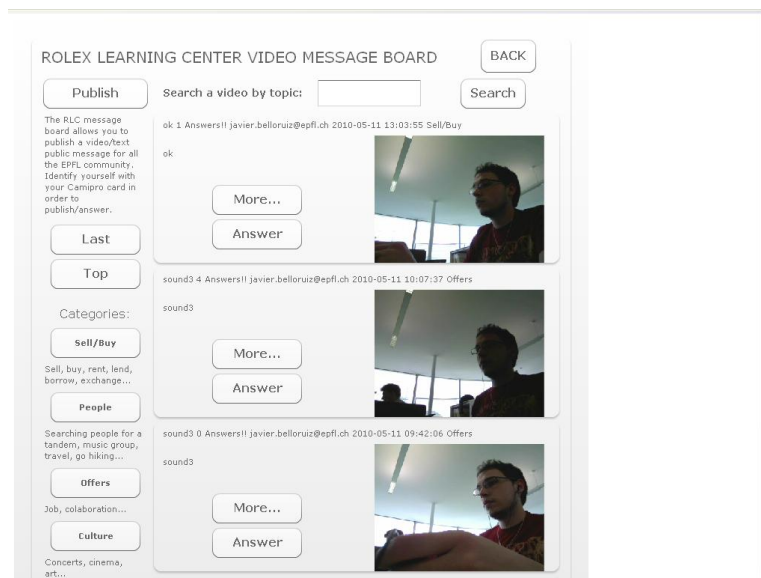


Figure 3.2: Modified prototype of the video message board application in white colors, trying to follow the design of the Rolex Learning Center

also necessary with this SCIPER number to access a LDAP Web Service in order to get the final information needed for the different applications as we can see on the schema (Figure 3.6).

We are going to explain in brief the procedure needed to access these different Web Services:

- SOAP

SOAP, originally defined as Simple Object Access Protocol, is a protocol specification for exchanging structured information in the implementation of Web Services in computer networks. It relies on eXtensible Markup Language (eXtensible Markup Language (XML)) as its message format, and usually relies on other Application Layer protocols (most notably Remote Procedure Call (RPC) and HTTP) for message negotiation and transmission. SOAP can form the foundation layer of a Web Services protocol stack, providing a basic messaging framework upon which Web Services can be built. This XML based protocol consists of three parts: an envelope (which defines what is in the message and how to process it), a set of encoding rules for expressing instances of application-defined data types, and a convention for representing procedure calls and responses.

Most of SOAP based Web Services have a configuration file called Web Services Description Language (WSDL) that make you able to know

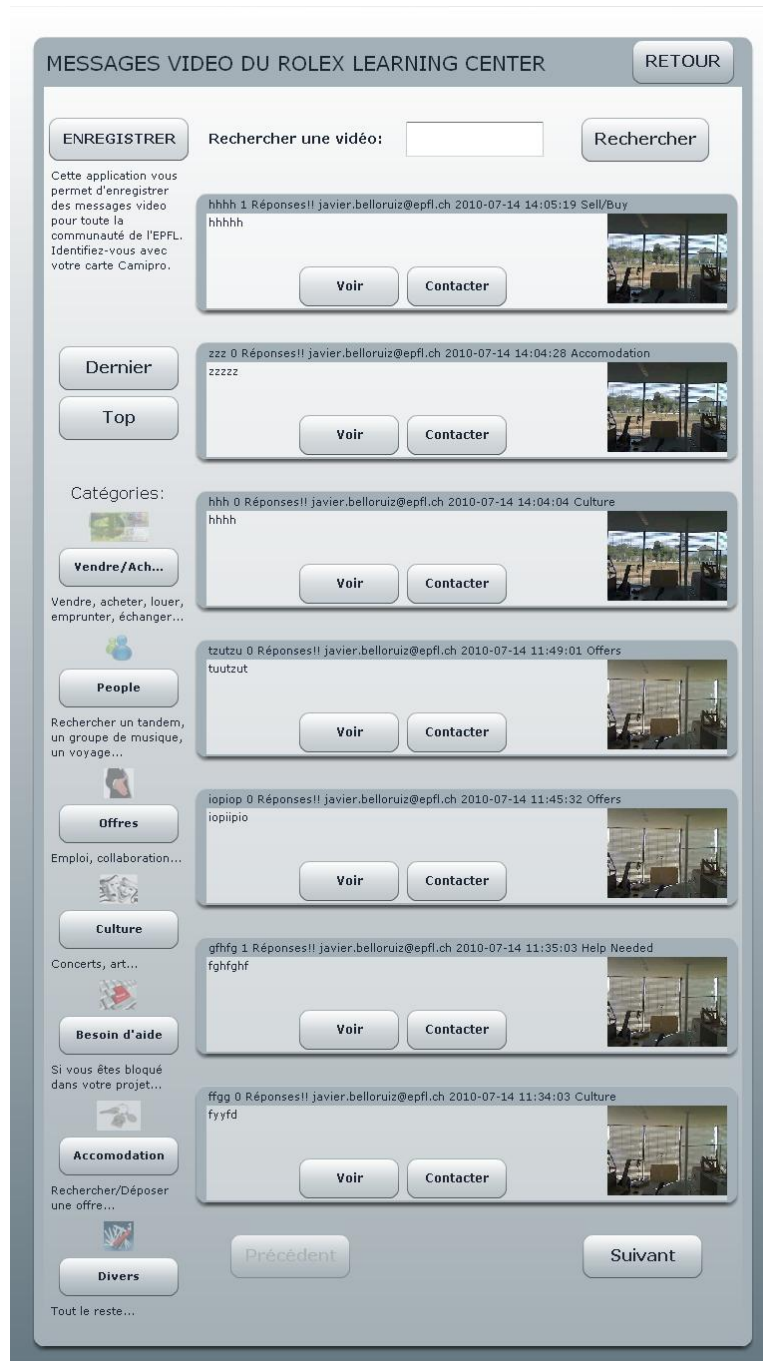


Figure 3.3: Final prototype of the video message board application with more videos per page and better quality colors following the design of the Rolex Learning Center

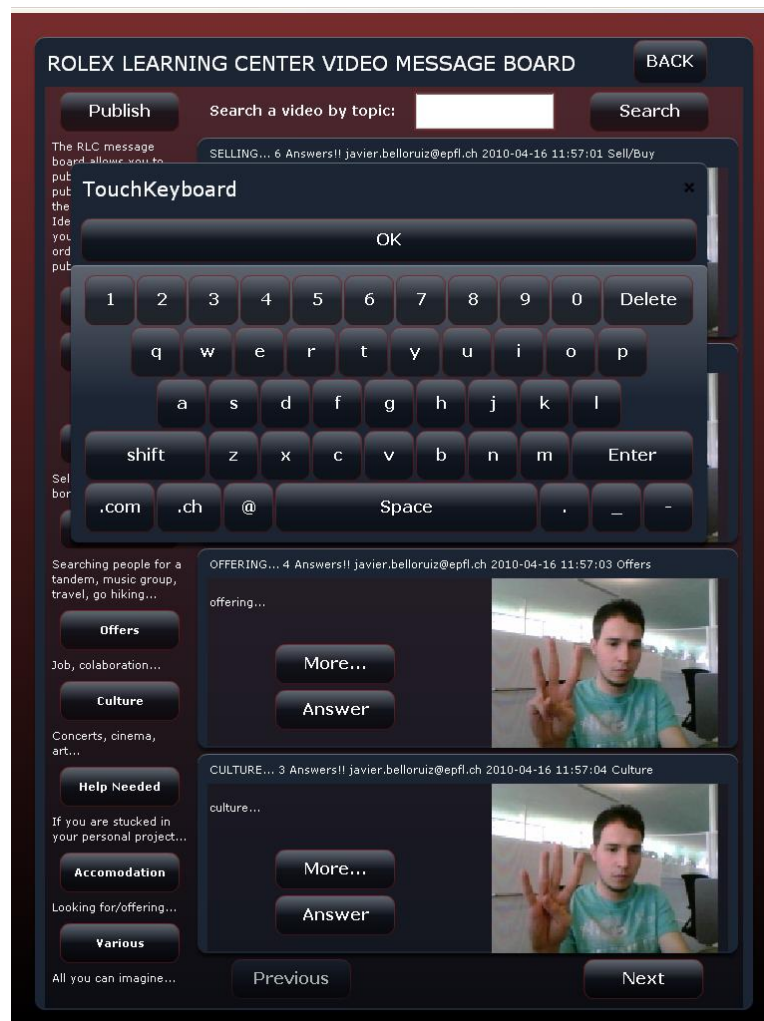


Figure 3.4: First prototype of the virtual touch keyboard

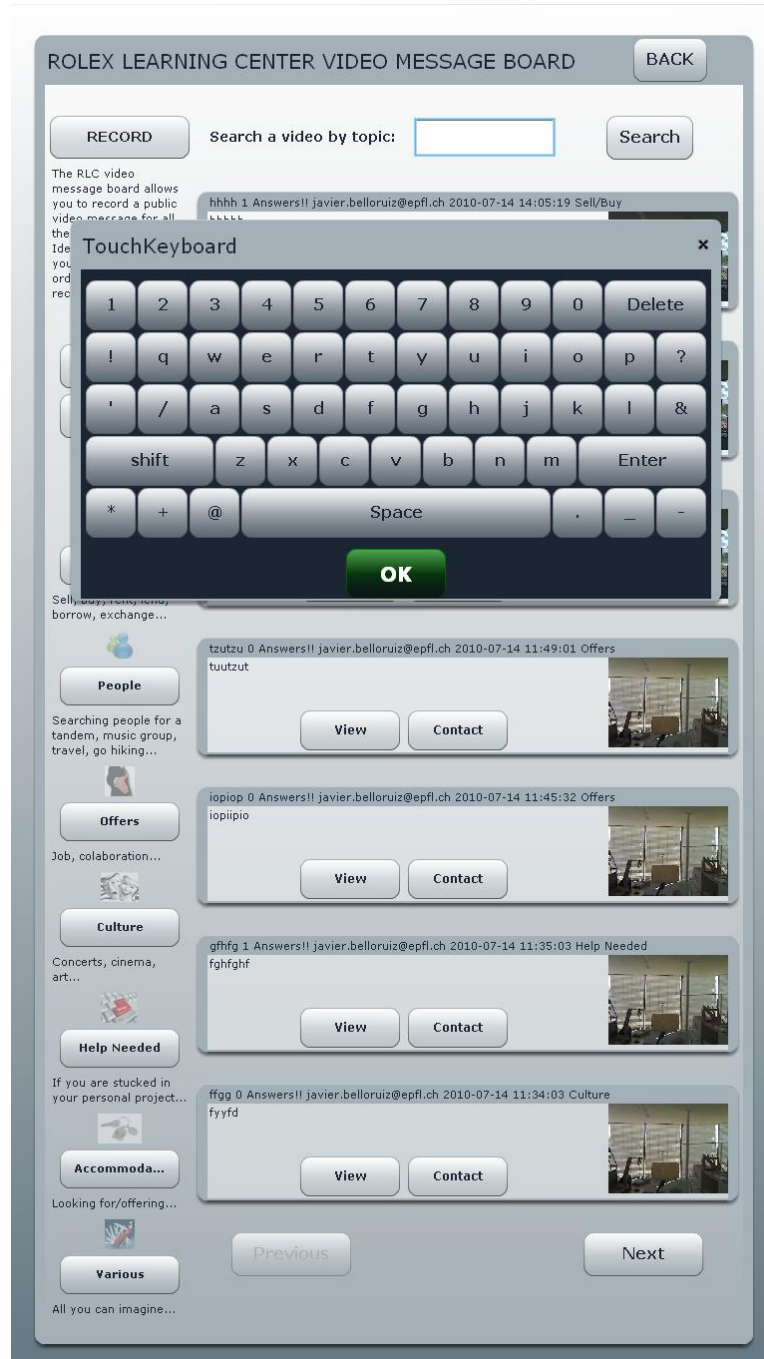


Figure 3.5: Extended prototype of the virtual touch keyboard

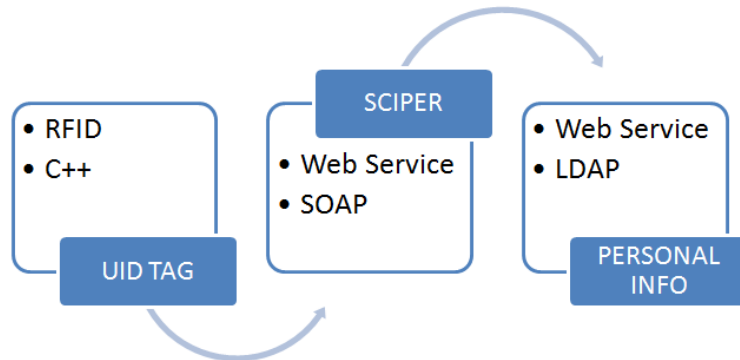


Figure 3.6: Process for identifying the user through the RFID reader and the Web Services

the structure and methods of the Web Service, however this particular service has no WSDL file and then we could not use the capability of the Adobe Flash Builder platform of building the manager classes for this service automatically from this file. Instead of that, the easiest approach in this case was using the support of SOAP provided by PHP [3] and connect the PHP file with ActionScript 3.0 as usual. The only restriction was that the IP address of the PC had to be sent to the administrator of the service in order to allow you access the service <http://dinfo.epfl.ch/Camipro> and use the "getSciper" method.

- LDAP

LDAP, the Lightweight Directory Access Protocol is an application protocol for querying and modifying data using directory services running over TCP/IP. A directory is a set of objects with attributes organized in a logical and hierarchical manner. A simple example is the telephone directory, which consists of a list of names (of either persons or organizations) organized alphabetically, with each name having an address and phone number associated with it.

For accessing the LDAP Web Service directory <ldap://ldap.epfl.ch/> was not necessary a permission for this IP address, however we did not know the structure of the directory in order to query the appropriate information. In order to discover this information was necessary to use a PHP program [1] for showing the whole tree structure (Figure 3.7) of the directory making us able to query the information about the user through different indexes.

```

/*
Array
(
    [count] => 2
    [0] => Array
        (
            [ou] => Array
                (
                    [count] => 2
                    [0] => Section d'informatique - Semestre printemps
                    [1] => IN-E
                )
            [0] => ou
            [count] => 1
            [dn] => cn=Javier Bello Ruiz,ou=in-e,ou=in-s,ou=etu,o=epfl,c=ch
        )
    [1] => Array
        (
            [ou] => Array
                (
                    [count] => 2
                    [0] => Recherche et appui pour la formation et ses technologies
                    [1] => CRAFT
                )
            [0] => ou
            [ou;lang-en] => Array
                (
                    [count] => 1
                    [0] => Pedagogical Research and Support
                )
            [1] => ou;lang-en
            [count] => 2
            [dn] => cn=Javier Bello Ruiz,ou=craft,ou=aa-vp,ou=aa,o=epfl,c=ch
        )
    )
)
*/
$r=$info[0][ou][0];
//Section d'informatique - Semestre printemps
$r=$info[0][ou][1];
//IN-E

```

Figure 3.7: LDAP directory structure

3.2.3 Persistence

In the previous chapter we introduced the initial entity-relationship diagram of the database. Later in this design we discovered the need of adding more tables to this database for giving support of the statistics, system logs and feedback of the applications, therefore the final tables implemented were:

```
CREATE TABLE Person
(
  sciper INTEGER NOT NULL PRIMARY KEY,
  email VARCHAR(30) NOT NULL UNIQUE,
  complete_name VARCHAR(45) NOT NULL
)
```

```
CREATE TABLE Category
(
  category_id VARCHAR(15) NOT NULL PRIMARY KEY,
  messages_number INTEGER
)
```

```
CREATE TABLE Message
(
  message_id VARCHAR(26) NOT NULL PRIMARY KEY,
  filename VARCHAR(30) NOT NULL,
  text VARCHAR(50),
  title VARCHAR(15),
  message_time DATETIME,
  answers_number INTEGER,
  category_id VARCHAR(15),
  user_id VARCHAR(30),
  FOREIGN KEY(category_id) REFERENCES Category(category_id),
  FOREIGN KEY(user_id) REFERENCES Person(email)
)
```

New table created for storing the number of people in each section/faculty/department using the video message board with a statistical purpose:

```
CREATE TABLE Section
(
  section_id VARCHAR(10) NOT NULL PRIMARY KEY,
  use_number INTEGER
)
```

)

New table created for storing the number of video messages played/recorded/contacted each day in the video message board with a statistical purpose:

```
CREATE TABLE Log
(
log_id DATETIME NOT NULL PRIMARY KEY,
answers_number INTEGER,
played_number INTEGER,
recorded_number INTEGER
)
```

New table created for storing the questions for giving support to the new today's question application:

```
CREATE TABLE Question
(
question_id DATETIME NOT NULL PRIMARY KEY,
yes_number INTEGER,
no_number INTEGER,
dk_number INTEGER,
text VARCHAR(500),
shown INTEGER,
shown_date DATETIME
)
```

Regarding the unique identifier of the message we chose to use the following representation with the SCIPER number belonging to the user that recorded the video message and the exact time and date of the recording:

RLC SCIPER_Date(YYYYMMDD)_Hour(HHMMSS) :
RLC_192993_20100416_115703

3.3 Software: decisions and components, used technologies

The following list contains the selected software during this phase of design in order to use it during the deployment:

- CurrPorts: this application was used in order to solve conflicts between the ports of the different servers used: Apache, MySQL, Flash Media Server, etc. CurrPorts displays the list of all currently opened TCP/IP and UDP ports on your local computer. For each port in the list, information about the process that opened the port is also displayed, including the process name, full path of the process, version information of the process (product name, file description, and so on), the time that the process was created, and the user that created it. In addition, CurrPorts allows you to close unwanted TCP connections, kill the process that opened the ports, and save the TCP/UDP ports information to HTML file, eXtensible Markup Language (XML) file, or to tab-delimited text file. CurrPorts also automatically mark with pink color suspicious TCP/UDP ports owned by unidentified applications (Applications without version information and icons).
- AVG Anti-Virus Free Edition 9.0: free antivirus software used to provide protection to the system.
- Dropbox 0.7.110: software used in order to share files between different computers in a cloud-based way for making backups of the project and to transfer code files for editing them in other computer.
- Adobe Flash Media Server: media server provided by Adobe for managing streams of audio and video through the Internet for making able Flash apps to record or play streams on the web.
- Adobe Flash Media Live Encoder 3.1 live audio and video capture software is a media encoder that streams audio and video in real time to Flash Media Server software or Flash Video Streaming Service (FVSS).
- Flash Player Plug-in 10.1.53.64: Adobe Flash Player is a cross-platform browser-based application runtime that delivers uncompromised viewing of expressive applications, content, and videos across screens and browsers. Flash Player 10.1 is optimized for high performance on mobile screens and designed to take advantage of native device capabilities, enabling richer and more immersive user experiences.
- Apache httpd 2.2.15 win32 x86 no-ssl: web server installed in order to make accessible the system via Internet, providing support for the server-side PHP files for managing the system calls and the MySQL queries [2].
- MySQL 5.1.45 win32: SQL based database server commonly used in web developing.
- Notepad++ 5.6.8: code files editor supporting several programming languages highlighting and other advanced features.

- PHP 5.3.2 win32 VC6 x86: server side language service commonly used in the Internet for managing system calls, MySQL queries and other features that HTML or other developing languages for web pages cannot do it by themselves.
- VLC 1.0.5 win32: VLC is a portable multimedia player, encoder, and streamer supporting many audio and video codecs and file formats as well as DVDs, VCDs, and various streaming protocols.
- Adobe Flash Pro CS5: software used in order to provide the system of some animations for the Flash platform
- Adobe Photoshop Elements 4.0: we used this software to produce the *.PNG images used in the different parts of the system like icons, standalone images, screensaver etc. We used free rights images of the Internet and we modified it in order to fit to our screens.
- Touch Screen drivers: UPDD v4.1.6R INPUTECH, ELO TOUCH-SYSTEMS Pointer Device Settings, control software of the drivers associated with the touch screen, in which we can adjust the calibration of the screen and the threshold of the signal strength in order to consider it a valid touch among other features.
- ZOTAC Motherboard Drivers Version: 83011, drivers for the motherboard including the rotation capabilities for the integrated NVIDIA GPU.
- Flash Builder 4/Flex Builder 3, software environment based on the Eclipse platform in which I programmed and debugged the main features of the system for the ActionScript 3.0 files and the MXML components for the interface.
- Realtek, control software of the drivers associated with the integrated sound card of the system necessary for adjusting some parameters of the audio recording/playing of the webcam.
- NVIDIA, latest drivers for the integrated NVIDIA graphics processor, used in terms of rotation of the screen to fit into the kiosk and providing graphics support for the system.

We will explain in depth in the User's manual and development guide appendix the use and configuration of some of this software.



Figure 3.8: Final chosen webcam with integrated microphone

3.4 Hardware: decisions and components, used technologies

The final chosen hardware components are the following:

PHILIPS Portable Speakers, model no. SBP1100/00, S/N: HS1A0947503396, impedance: 0.8 ohms, speaker power: 2x1W. This model were selected because it has an integrated amplifier providing better sound than a previous model tested.

Microsoft LifeCam Show, part number: RLA-00001. This webcam has a built-in unidirectional microphone with acoustic noise cancellation and echo cancellation. It also has a 2.0 megapixel sensor with 5X digital zoom and a wide-angle lens of 71 degree and a focus depth of field from 21" to 60". It was selected because of the ultra-thin mobile design that allowed us an easier integration in the kiosk totem (Figure 3.8).

Motherboard ZOTAC ION ATX-A-E mini-ITX.
 Chipset: NVIDIA MCP7A-ION Series. NVIDIA nView multi-display (up to 2 monitors). Size: Mini-ITX form factor of 6.69 inch x 6.69 inch (171 mm x 171 mm).
 Microprocessor: Intel ATOM N330 (Dual-Core) CPU.

RAM CORSAIR XMS2 4GB(2x2GB) DDR2.

RFIDGeek TRF796x Multiple-Protocol RFID Reader/Writer, Univelop Comp. Tech. LLC:

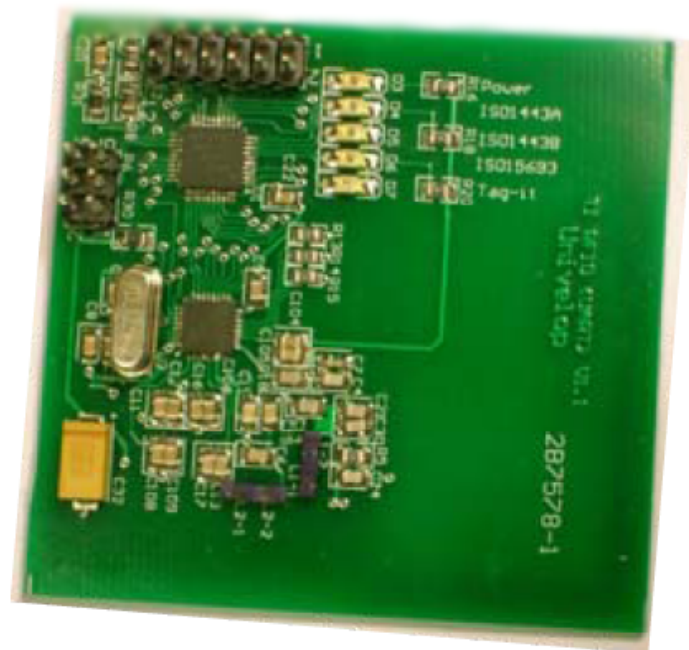


Figure 3.9: RFID UART reader

The TRF796x is an integrated analog front end and data framing system for a 13.56 MHz RFID reader system. Built-in programming options make it suitable for a wide range of applications both in proximity and vicinity RFID systems. The reader is configured by selecting the desired protocol in the control registers. Direct access to all control registers allows fine tuning of various reader parameters as needed. Supports multiple protocols ISO-15693, ISO-14443A, ISO-14443B(Philips MiFare RFID) tags and Texas Instruments Tag-it standard.

The operation frequency of the reader is 13.56 Mhz and has less than 100ms decoding time. It's composed by a TTL interface with an optional USB/RS232 interface with a built-in transceiver antenna with maximum effective distance up to 10 cm and 5 built-in LEDs to indicate which type of protocol card you are using with a final size of 58 mm X 56mm.

We are going to take a look to the different components of RFID Reader in the following figures: UART Multiple-protocol RFID Reader/Writer (Figure 3.9) with the built-in transceiver antenna and USB/RS232 interface board(2.1.4) to be used with USB (Figure 3.10).

For the connection between USB/RS232 Interface board and RFID

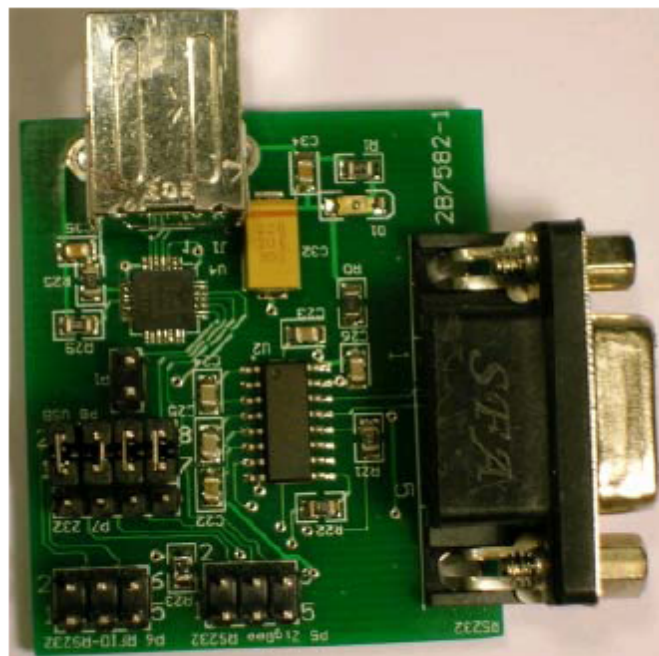


Figure 3.10: RFID UART interface

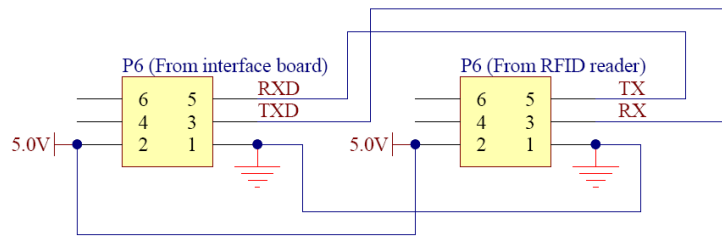


Figure 3.11: Connection between the RFID UART reader and the RFID interface for the PC

Reader/Writer you can use a 6-pin cable following the schema on Figure 3.11.

We will talk about the ISO-15693 protocol commands used for reading the CAMIPRO cards on the User's manual and development Guide appendix.

Chapter 4

Deployment, system test and evaluation

In this chapter we are going to describe the deployment of the system and the different test carried out to ensure a reliable application. We are going to list the main detected problems and how we solved them.

4.1 Final deployed design

In this section we have a review of the different built parts of the totem as well as the final interface design with a brief description of each figure.

In Figure 4.1 we can see the arrangement of hardware devices used in the system, placed in the built totem for the interactive kiosk.

In next Figure 4.2 we have a general view of the interactive kiosk placed in its final location in Rolex Learning Center.

In next Figure 4.3 we have a general view of the closed back of the totem for the interactive kiosk.

Following Figures 4.4, 4.5, 4.6, 4.7, 4.8, 4.9, 4.10 and 4.11 we have the final GUI of the interactive kiosk, with the main screen of each app of the system.

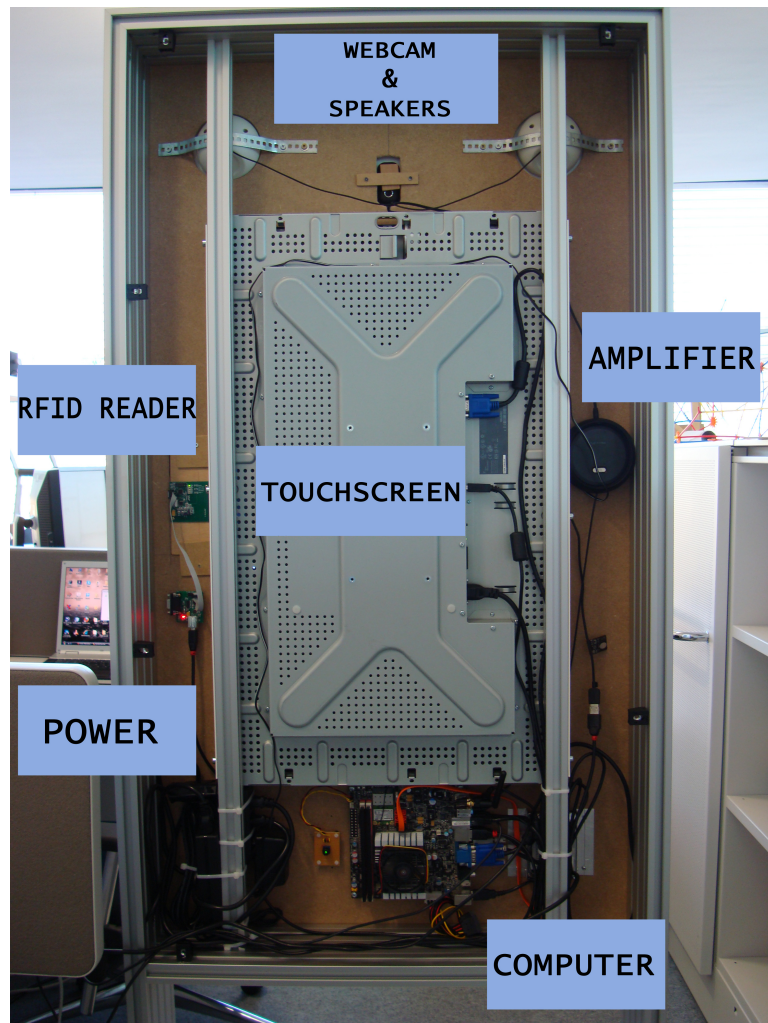


Figure 4.1: Final hardware arrangement in the kiosk's totem back



Figure 4.2: Final built and working prototype of the Interactive Kiosk for the Rolex Learning Center



Figure 4.3: Totem back detail



INTERACTIVE KIOSK



Rolex Learning Center

Figure 4.4: Main screen

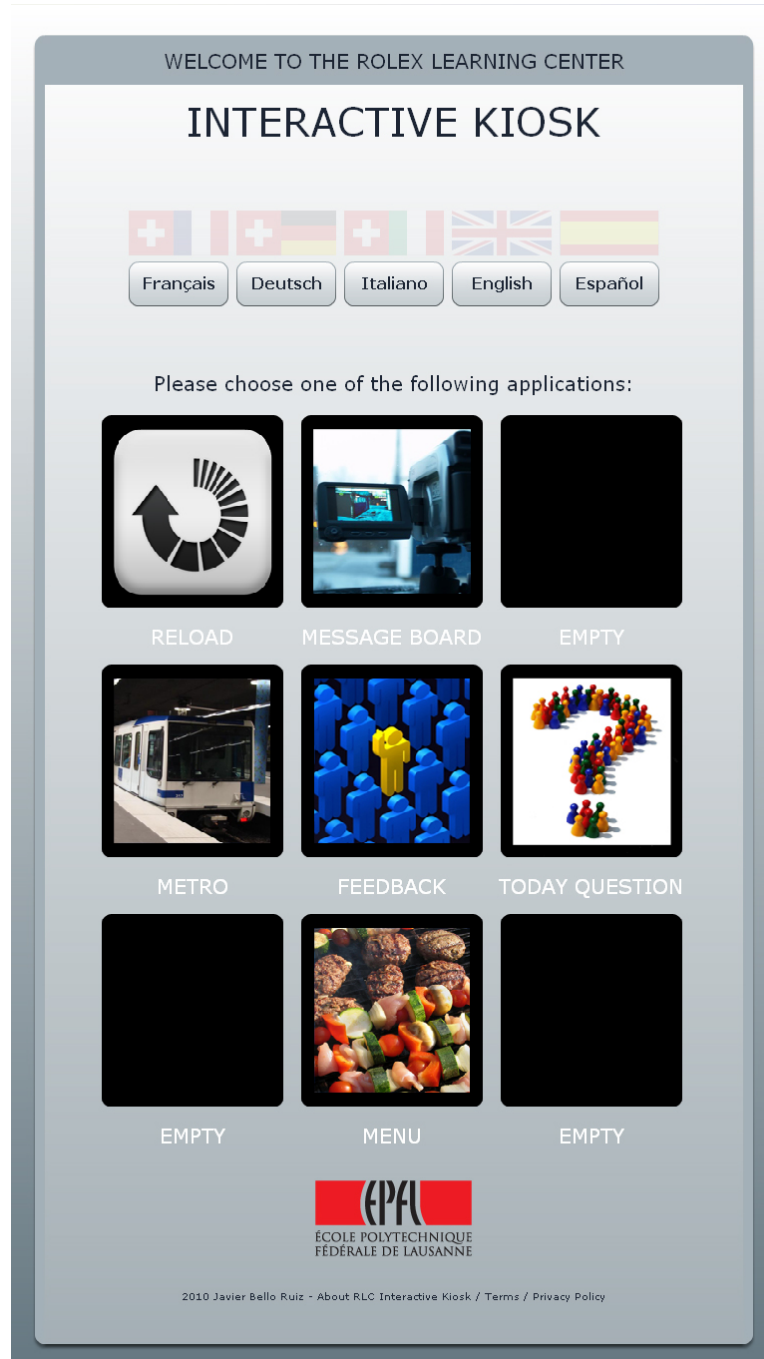


Figure 4.5: Apps screen

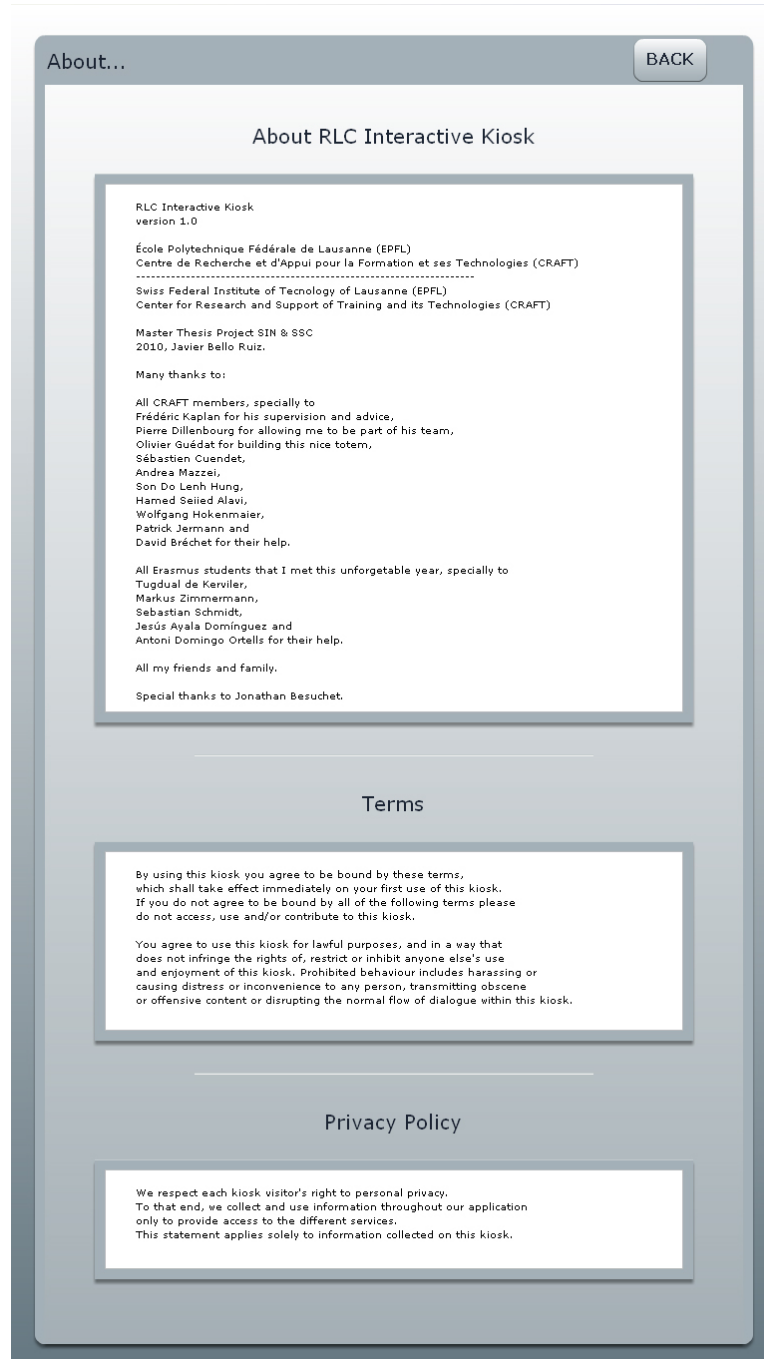


Figure 4.6: About, terms and privacy policy

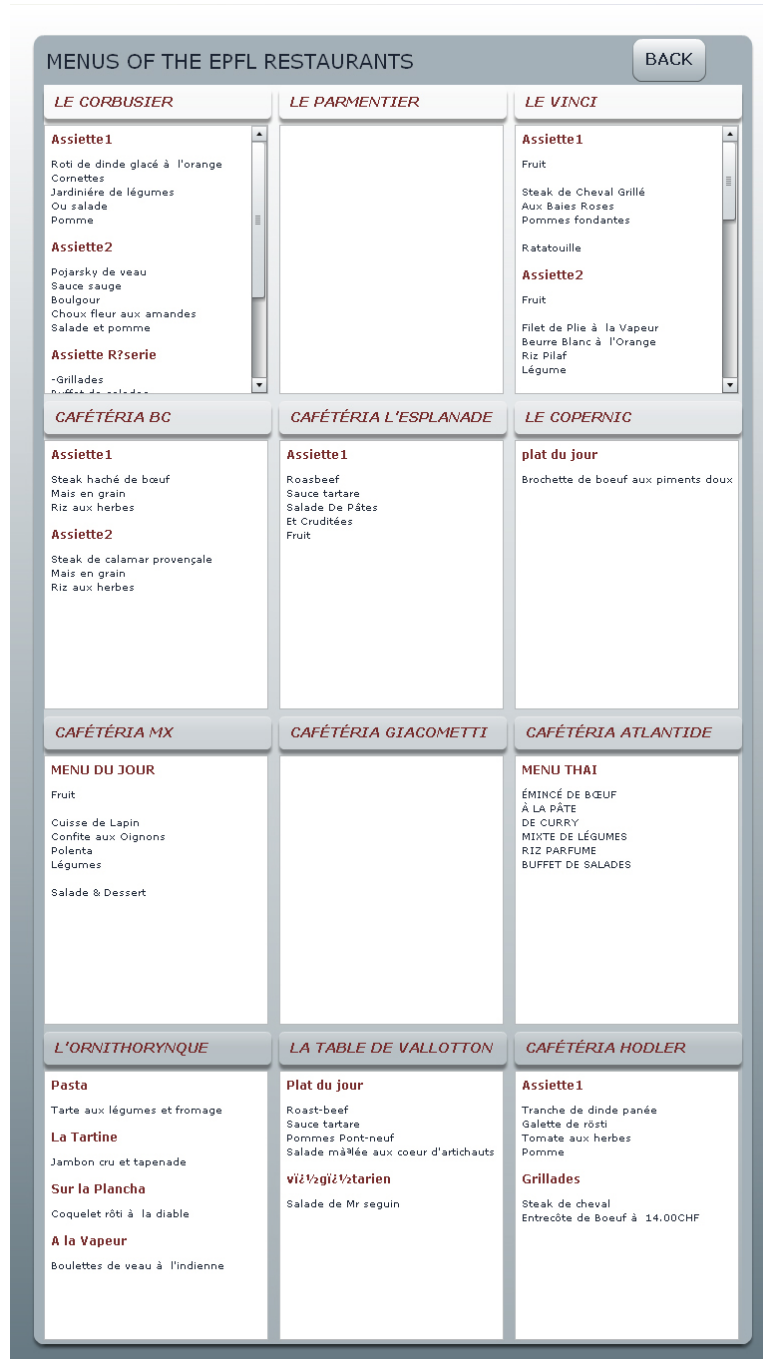


Figure 4.7: Restaurant menus app

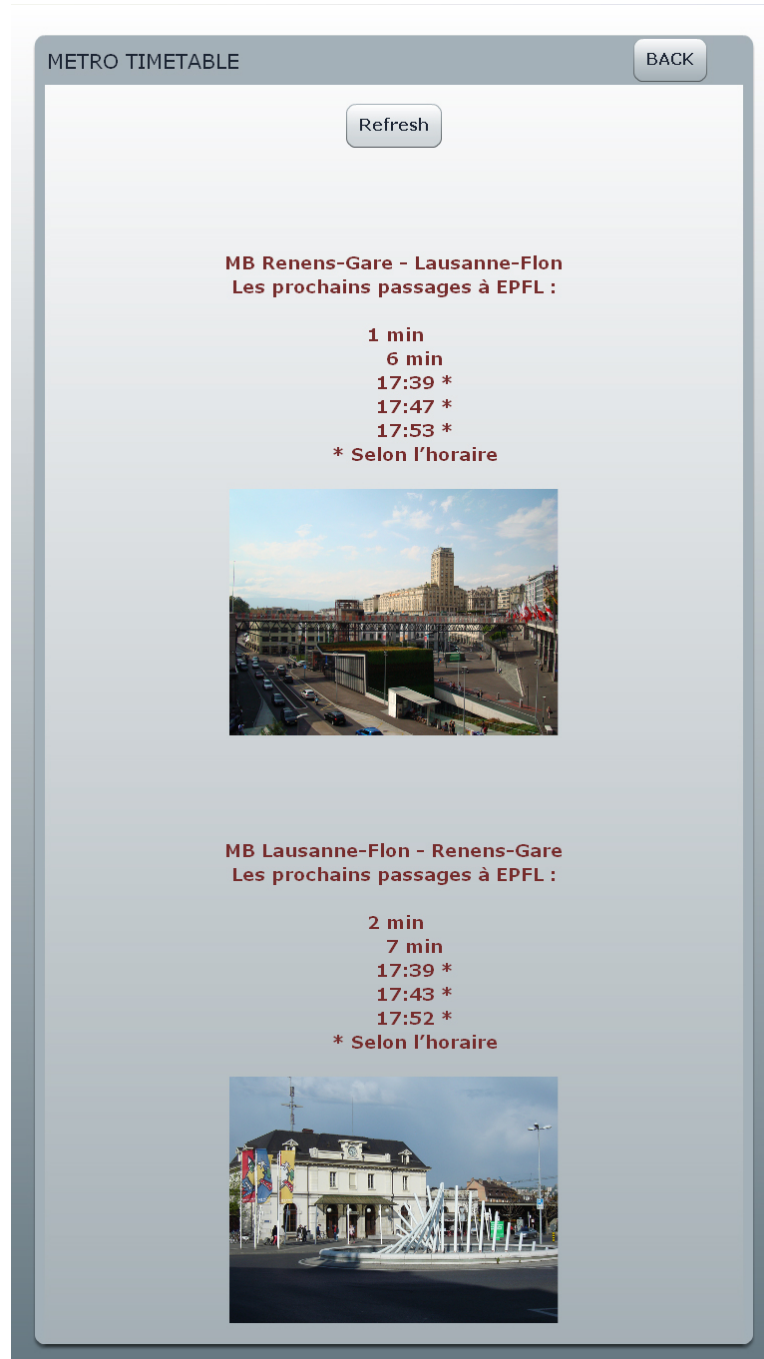


Figure 4.8: Metro timetable app

FEEDBACK ABOUT THE APPLICATIONS

BACK

Please give us your feedback about the content of the kiosk:
problems, bugs, recommendations, things that are not clear, issues...

1	2	3	4	5	6	7	8	9	0	Delete	
!	q	w	e	r	t	y	u	i	o	p	?
'	/	a	s	d	f	g	h	j	k	l	&
shift		z	x	c	v	b	n	m	Enter		
*	+	@	Space					.	-	-	

SEND COMMENT

Figure 4.9: Feedback app

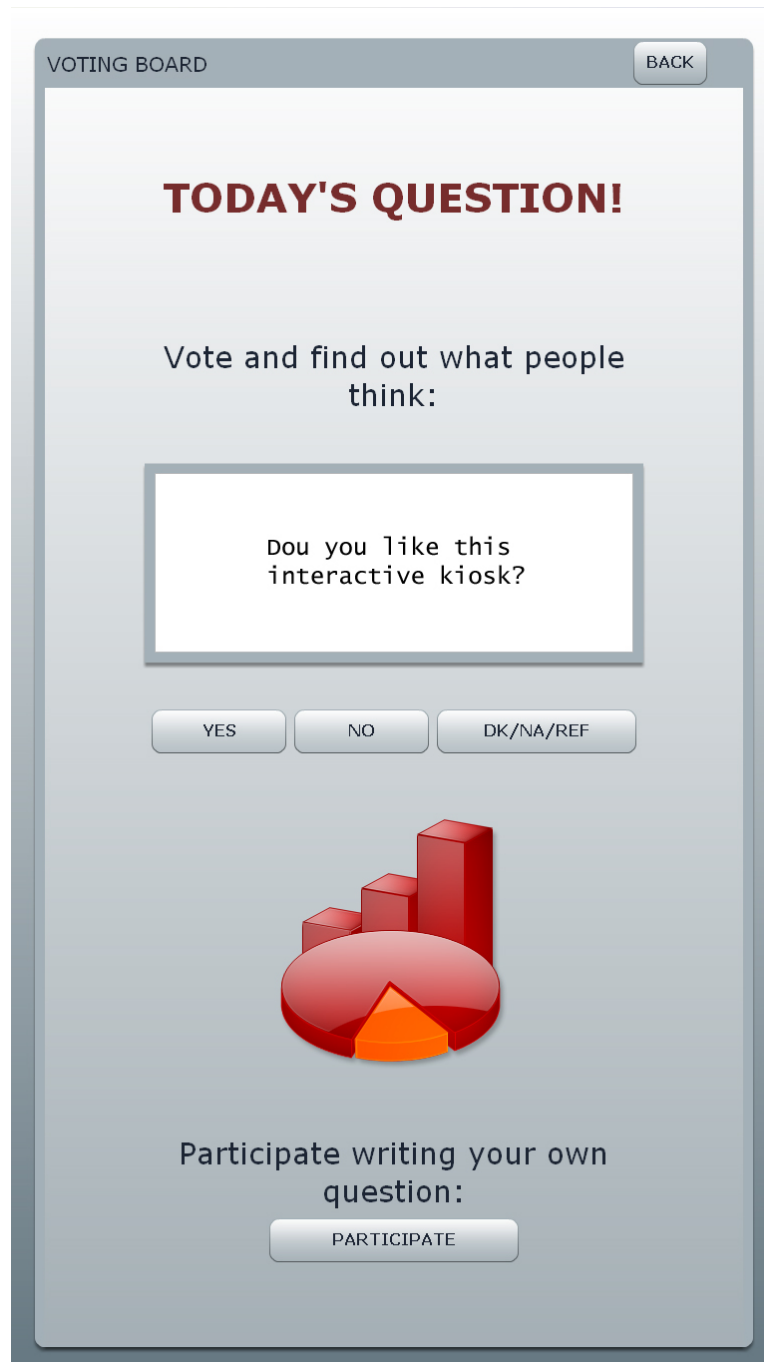


Figure 4.10: Today's question app

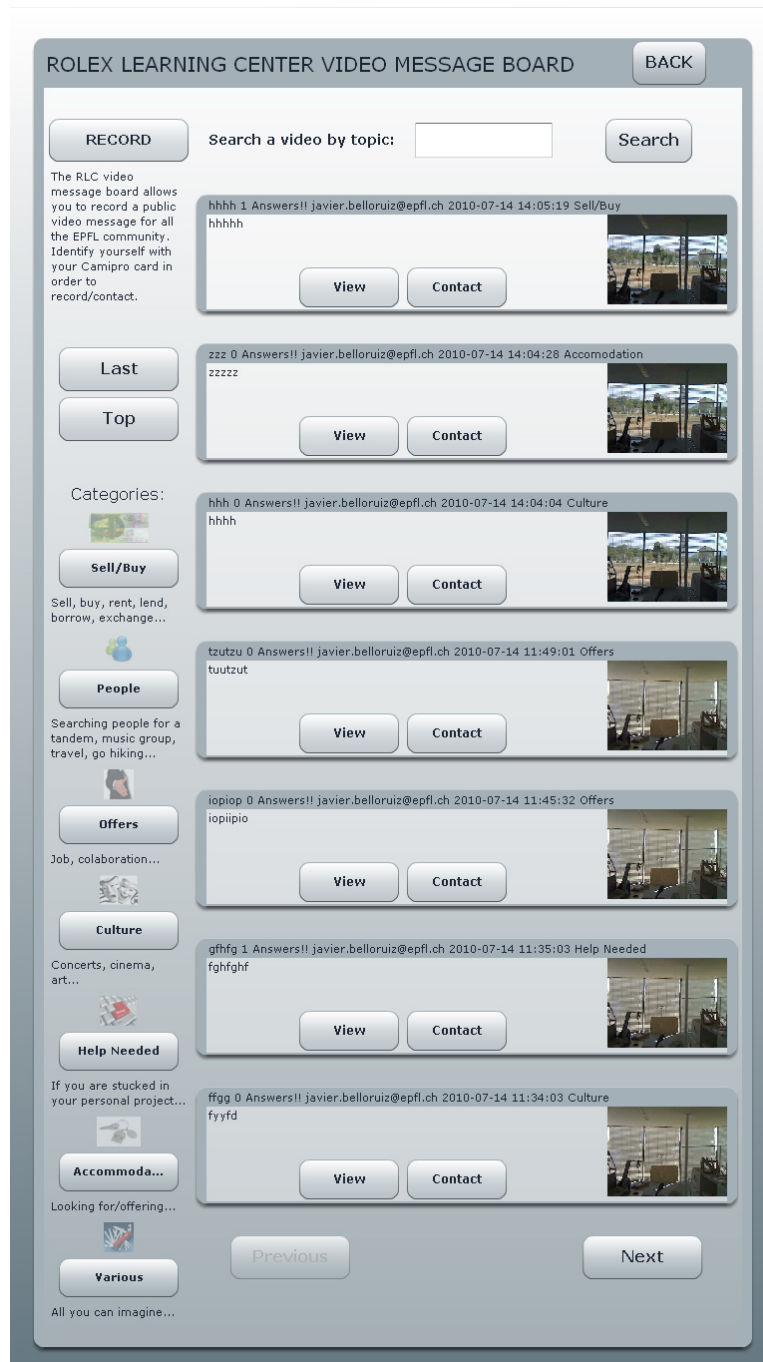


Figure 4.11: Video message board app

4.2 Detected problems

After finishing the building process of the kiosk, we detected some problems in the initial tests of the whole system working together.

The first one and more important happened when running the system for a long time: flash player was freezing for some time the web browser. This problem was performed by two different things: the first one was produced because, in the Screen Saver component, two videos were looping once and again and both were launching the same event and trying to capture it later, making the system instable. In the other part we did not notice this problem before, because we did not run the application for more than two hours and the flash player plug-in was not reporting the bug.

Since the flash player plug-in and Firefox were updated to the next version, the problem became more remarkable because both videos interfering one into the another were making the system instable for more than 45 seconds. This 45 seconds were the limit on new Firefox 3.6 to wait before stopping a non-responsive plug-in, then making flash player crash. We took an easy solution, just removing that event throwing and stop looping the videos.

With this updated version of the Flash Player plug-in, some issues were produced by the fact that the FLVPlayback 2.5 component for playing videos was designed in principle for using it with Flash CS4 Professional with plug-in version 10.0 and supported by Flex 3 and its SDK. When we upgrade the system from Flex SDK 3.5 to 4.0 and later the flash player plug-in from 10.0 to 10.1 the program was throwing unexpected events making it instable and we had to handle these new events.

Another problem with these videos was that, as we were waiting the user interaction to exit the screensaver, I mistook the place of the handling function for this event placing it in the stage occupied by this component and not in the whole screen, making the videos play without being shown in the screensaver. The solution for this problem was placing the function that close the video streams in an outer component of the application.

At the end of the deployment we had some problems with the video quality: after buying a new camera with better resolution and that fitted very well in the design, in the end we realized that recording through the Flash Media Server added a delay between the video and the sound that was increased by the better resolution, making a better quality camera provide worse results in terms of video quality. Reducing the quality in the camera properties allowed us to reduce that delay and after trying several configurations we managed to had a not bad quality video. We asked the opinion about the final video quality in the use study with good results.

4.3 Maintenance and management

Once the whole system is working together, one of the most important things is how to maintain the system working properly during its lifetime. For that, as the device is placed into the building we must have a way to access remotely the inside PC for solving any kind of future problems or for improving the system with new features.

In order to do that we made able the access through Windows Remote Desktop configured with the administrator account:

Admin: Javier Bello Ruiz
Password: RLCpfc2010
I.P.: 128.178.13.216

In other side, for managing the email account associated with the kiosk, from where we send the contact emails of the video message board, we provided access to the account with the following data in:

<http://webmail.epfl.ch/>
User: rlc.kiosk@epfl.ch
Nick name: mail_rlckiosk
Password: Mailrlckiosk2010

At first, we considered to have an administration console to be able to moderate the different video messages and have access to the logs of the system. We thought of including features like being able to read the feedback of the people, have charts with the use of the kiosk and delete inappropriate videos plus sending an email to the one who posted, warning him not to post messages of this kind.

But due to restrictions with the Flash Media Server, that in the free release has a limited capacity of 10 open video streams, we could not have both applications running without interfering between them, so we decided not to go far into this app and use the remote desktop to access the videos, database, logs, etc. directly and remove the inappropriate content with a simple PHP file.

The steps of the protocol for testing the system are: Switch on. Wait for the system to initialize. Check if Flash Media Server is working correctly. Check the calibration of the touch screen. Check the quality of the audiovisual components: Realtek, LifeCam, Flash settings. Initialize the Remote Desktop in case of a problem. Log off the system on the portable computer. Restart the main computer.

4.4 Use study: experimental results, qualitative analysis

After running the use study with ten different people we realized about two important things:

First of all, the interaction with this type of screen may be difficult in some aspects. Regarding the interaction with the camera and the virtual keyboard we found two different issues. One of them is related to the different height of people: for some people the camera is too high and for others the keyboard is too low and viceversa. One possible software solution is to use face recognition features for arranging the items into the screen in different ways depending on the person. Other hardware possibility is to make the screen physically adjustable, or the camera itself by rearranging it into a side position of the screen instead of being on top. The other issue is the touchscreen calibration regarding the buttons/keyboard as we have the same problem with people height; different people looking at the screen that is vertical to the ground makes them to see the buttons in different angle, that if the drivers of calibration are not accurate enough made people touch keys that they don't want to. There are three possible solutions for this problem, similar face recognition to place the items into the screen in different ways, make the buttons more spread and bigger to avoid misunderstandings, or finally had a better calibration of the screen regarding all these aspects. As the screen is PCT, the placement of a glass for protection made also a bit more difficult the calibration in terms of sensibility and accuracy. One easy solution can be point the device in an angle that should be easy for people to touch the keys.

The other important thing is that in this type of building with such open corridors etc. people is going through and not standing in the same place for a long time, except for resting in the puffs, that is making more interesting the applications of metro and menu that the video messages one. One possible solution is to reduce the number of clicks per action in this application, but as result of the use study makes the metro app the most useful one, other approach is only to focus in small apps just click and get the information.

You can find the followed use study protocol in the appendix named Use study protocol and check the different phases of it for understanding the following conclusions obtained after the analysis of the data collected.

After watching carefully the videos of the use study we extracted some important information: Nobody has problems with the feedback application, is easy and clear to send a comment. However taking into account the

ergonomics of the kiosk, the place where is placed the virtual keyboard is not comfortable for many users, because they bend the back in order to reach the keys without losing the perspective or for avoiding mistaking keys due also to the angle that people look at the keyboard. Possible solutions are moving the position or making the keyboard as big as nobody has problems with it. Another issue seems to be the no possibility of editing in the middle of a sentence that makes people or delete until reach the error or event worst in the text inputs trying to fix the cursor and deleting the whole sentence.

In the task of changing the language into your mother tongue and playing a video of accommodation category they seem not to have problems, they realize very soon where to change the language and they go straight to the message board and look for the accommodation section. Only one person did not realize about the fact that the videos belong to a category.

In the task of sending a question there were more hesitation and doubts than in the previous ones: two people hesitate where to participate with their own question but once they made the mistake they realize quickly where to go. Some common doubts were that in no place is written that the question is appearing other day, so some of them expected to see their question written in the answering panel. Three of them bend to write and two of them use two fingers in order to write faster and more accurate. One of them tried to edit the sentence in the middle and is not possible. Regarding the questions, they introduced a correct yes or no question 9 times out of 10 people, but this accuracy was due to the fact that in the free exploration of the kiosk most of them saw a wrong question, so the conclusion is that or either we have to make more clear that the question is yes or no question or making them able to fill the answers.

For the case of the task of recording a video message we structured the analysis procedure into steps in order to see in which ones people has more difficulties and we collected the result into the following Figures (4.12, 4.13, 4.14, 4.15, 4.16, 4.17, 4.18, 4.19, 4.20, 4.21, 4.22).

Some patterns observed during the free exploration of the kiosk are:

- They try to press the image on the metro timetable app.
- They always press the play button in the video message board instead of pressing the miniature of the video which is also possible.
- There is not clear relation between category and video in the video message board.
- They usually bend the back to write properly.

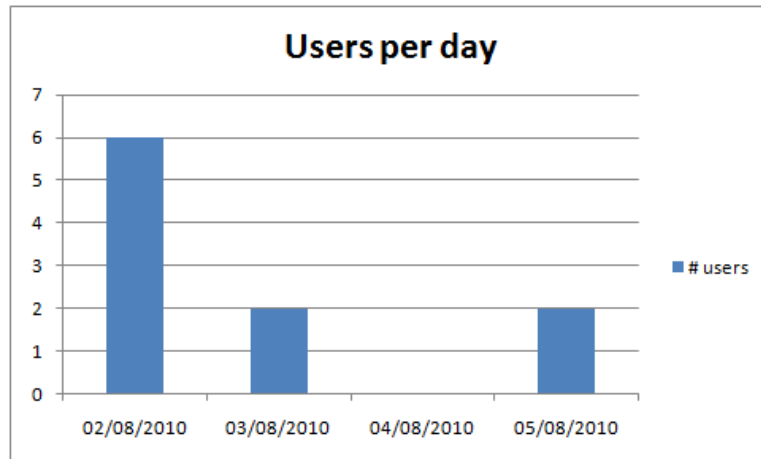


Figure 4.12: Number of users per day in the use study: probable impact in the logs of the video message board

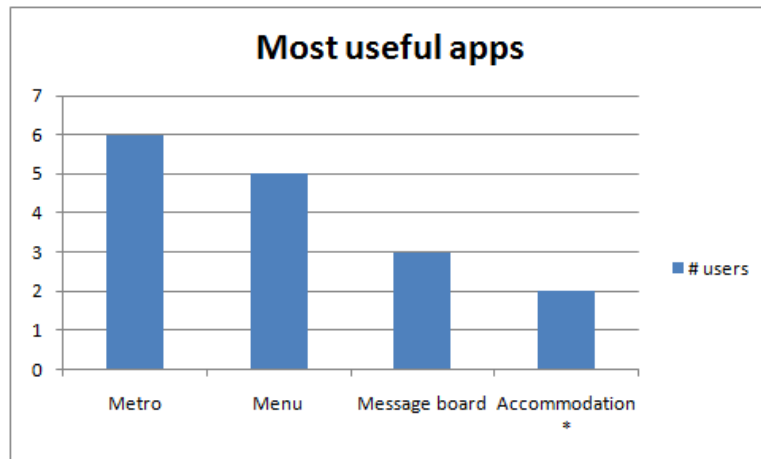


Figure 4.13: Most useful applications according to the use study questionnaires

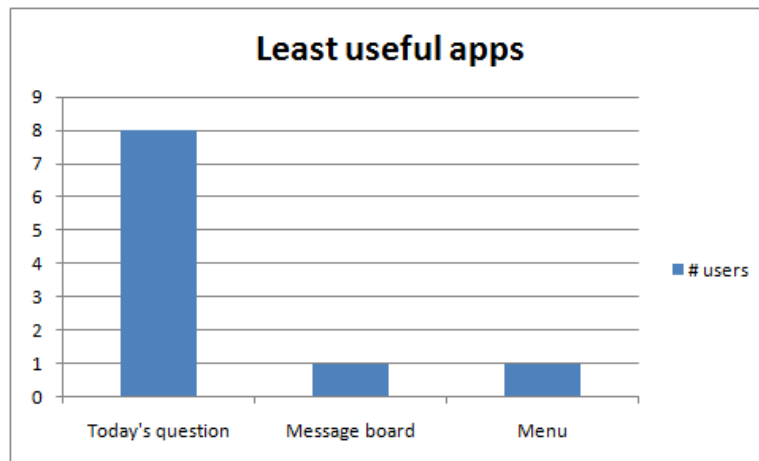


Figure 4.14: Least useful applications according to the use study questionnaires

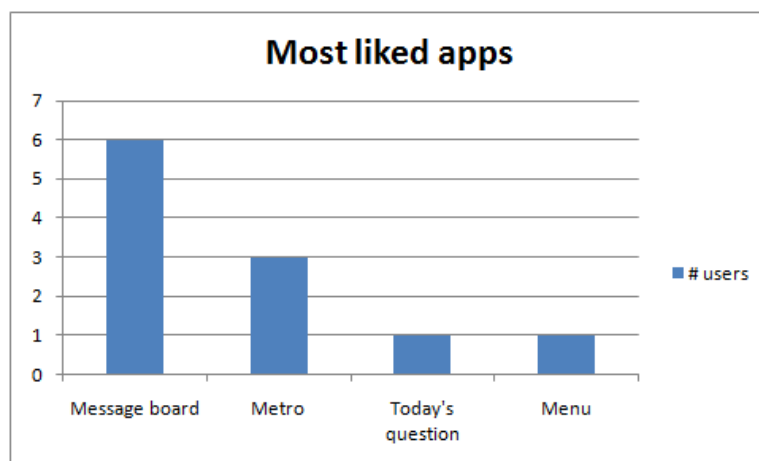


Figure 4.15: Most liked applications according to the use study questionnaires

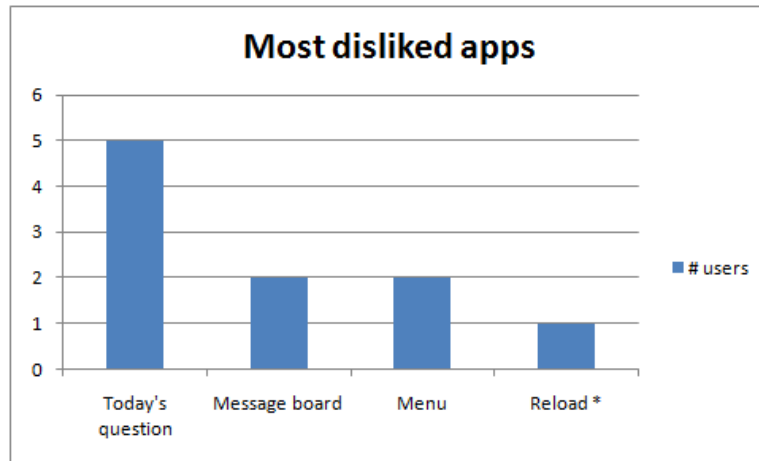


Figure 4.16: Most disliked applications according to the use study questionnaires

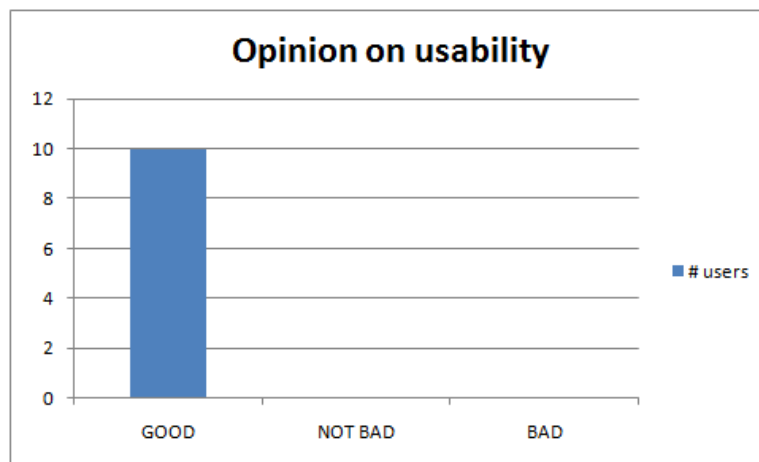


Figure 4.17: Opinion on the usability of the system

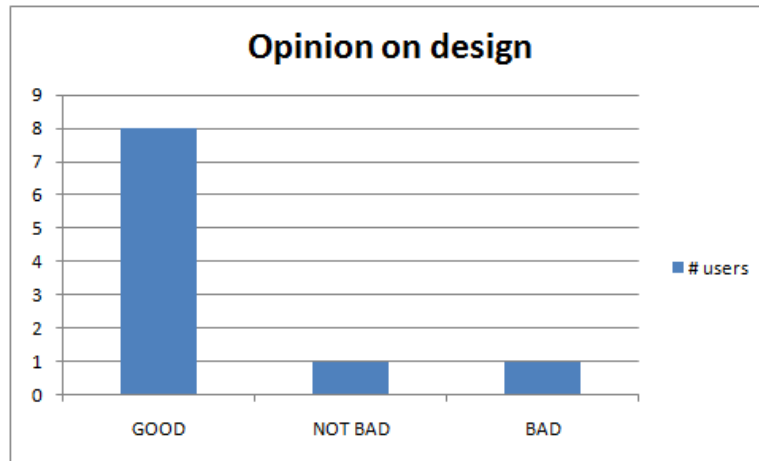


Figure 4.18: Opinion on the interface and device design

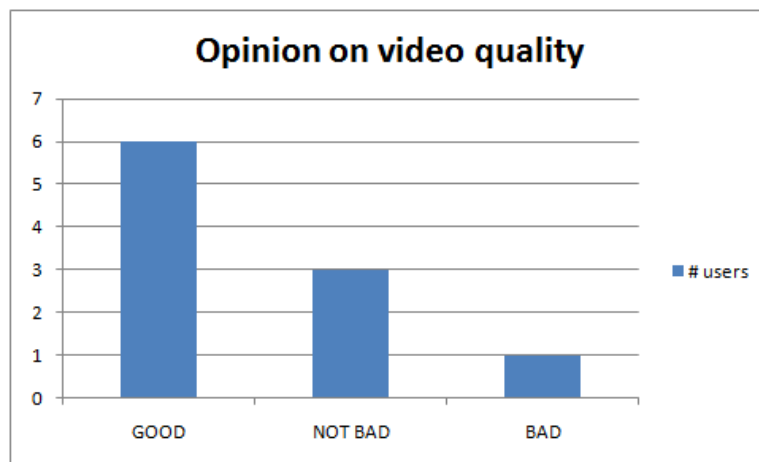


Figure 4.19: Opinion on the audiovisual quality in the video message board

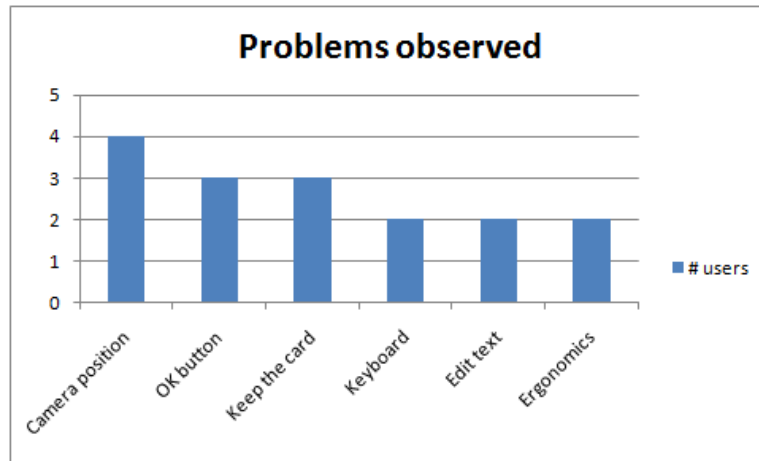


Figure 4.20: Problems observed during the free exploration of the kiosk

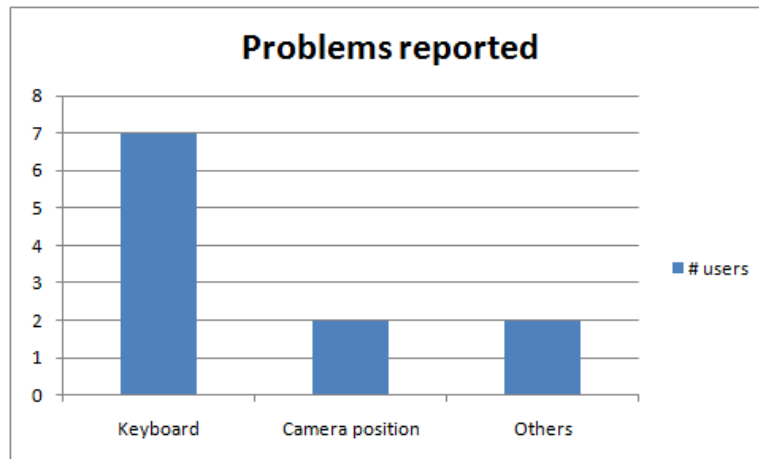


Figure 4.21: Problems reported by the users during the use study of the kiosk

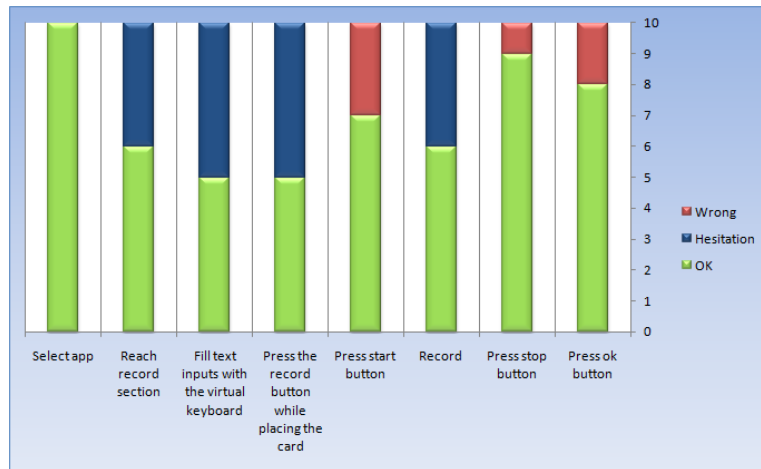


Figure 4.22: Evaluation of the usability in the task steps of recording a video message on the video message board during the use study of the kiosk

- There is a keyboard accuracy problem.
- There is a problem of accuracy with the OK button.
- They try to press the list of menus on the restaurant menu app.
- The question of the day usually does not make sense.

Chapter 5

Conclusions

In this final chapter we are going to sum up the most interesting things learned during this project, the conclusions obtained from the results and the analysis of future improvements of the system and further applications.

5.1 Personal review

For my personal experience, this project was very fulfilling and challenging. I learned a lot about the technologies involving this thesis and it was really amazing to see grow up all the development at the end of the project. On the other side I discovered that a good planning always leads to good results and it is one of the points that I have to improve for further projects. I enjoyed very much analyzing the problem and finding solutions for each of the necessary things needed for building an interactive kiosk and I think I succeeded on providing a final end system capable of being useful for people in Rolex Learning Center. I'm happy too for knowing that improvements can be made and other people can continue working on future prototypes. Regarding the scientific process followed for analyzing the problem and the results we discovered that in this kind of devices that a lot of different people are going to use it, it's really important to take care about the human details involving the design of the kiosk because as we saw during the use study every person has a different way to use the device and what is really amazing for one person can be a total waste of time for other.

5.2 More relevant aspects

At the end, we discovered that this kind of prototypes depend very much on people using it, and all the data we collected with this very first prototype leads to build better new ones, trying to solve the problems faced during the real usage of the kiosk and developing new solutions for everyday needs of people at Rolex Learning Center. We obtained positive results as the device is being used by the people on the building and is helping them with this applications we developed for everyday use.

5.3 Improvements, further applications

Looking at the obtained results, we can think of a lot of improvements for the kiosk in terms of ergonomics and user interface, it's necessary to reach a point in which the usability and the contents go together to provide useful applications and innovative ones related with the amazing environment of the building.

For the developed applications we can think about the following improvements:

Metro timetable app, as this is one of the most liked apps we can think about making better the service, including the buses that stop at the EPFL. In terms of making more reliable the application, we should ask the company for the different real time timetables instead of parsing the webpage.

Video message board app, as we discovered that people is not making a clear association between the categories and the videos, we can provide an interface that make the users select a category before looking the corresponding videos. We can also add a fun videos category or make an application of an interactive TV at Rolex Learning Center promoting an EPFL TV kiosk with features like editing video, face recognition, zooming, gesture recognition and subtitles.

Restaurant menus app, in this application the users have no problem to identify correctly the different menus of the restaurants but an improvement on the interface should be the next step.

Today's question app, as people are having troubles in writing a yes-or-no question and they find the application useless, we can think in other approach for this type of application that they can also edit the possible answers or participate in some way on Internet with an opinion webpage where you can consult the previous questions and results.

Feedback app, it is absolutely necessary to keep this app as we think that you can ever improve the services provided for the students and there are always worth new ideas for this kind of devices.

In terms of the design of the interface and of the physical device itself, we can think about the following improvements:

We need another approach for the virtual keyboard and for the placement of the camera, both are ergonomics issues and we can think about software solutions or physical solutions: face recognition to rearrange items in the screen as we said in previous chapter or a new design of the totem. In terms of the design of the interface it will be useful to have a designer in order to make more accessible and usable the interface and more clear to the user. A physical keyboard can be another approach.

Finally we can set up a web portal for people interested in the device, in order to provide them the appropriate tools, SDK or API, for developing applications for the device. In next iterations of the building process of the prototype, the physical location of the device in terms of light, surrounding sounds, etc. has to be more studied.

Future applications should consider again the students suggestions for providing them more services that they are demanding like the bike renting at the EPFL or other information. The future of having several kiosks working in a cloud based way offers the possibility of developing some applications in a collaborative way that all the kiosks can be participating at the same time.

5.4 Qualitative analysis of logs and feedback

These logs have been acquired between the 15/07/2010 and the 15/08/2010, as we can see analyzing Figure 5.1 there is an increase of the recorded videos on 02/08/2010 (6 people), 03/08/2010 (2 people) and 05/08/2010 (2 people) with the users participating on the use study. We can see too, two periods of inactivity (17/07/2010-18/07/2010 and 24/07/2010-01/08/2010) when the interactive kiosk was switched off and removed from the place due to a film and photography shooting in the building. Taking into account these singular events, we can anyway have a positive analysis of this data.

We can see that, as many videos we have recorded the increase of the played ones is increasing too, but as we see there is high interest in playing others videos than in recording new ones or definitely on asking/contacting somebody's video. But is a positive result to see that the played videos keep an interesting rate all over the time the device was placed. That means that despite of the fact that people is not using the possibility of asking the videos

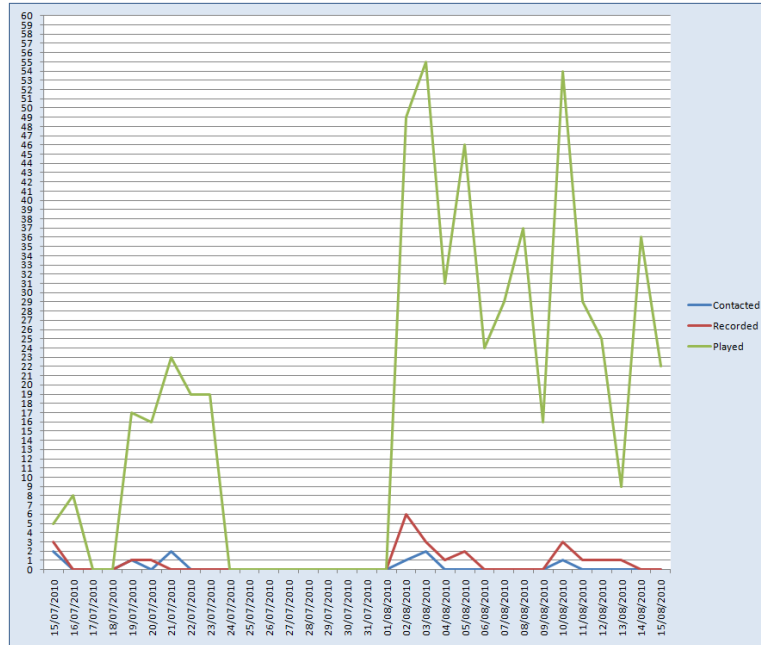


Figure 5.1: Logs of the video message board application: played, recorded and contacted number of videos in the period from 15/07/2010 to 15/08/2010

or contacting the person who recorded them, they are feeling like using the application for playing videos. And after making the use study we saw that as many videos we recorded, people were more enthusiastic on recording new ones.

I think that based on these results, the application is going on as we expected with a hopefully average use.

Looking at the comments that people provided us in the feedback application we can organize them into the following categories:

- Nonsense: 13 messages have not really meaning about the interactive kiosk.
- Good: 14 messages were giving good comments on the kiosk for being clear, easy to use and having useful applications like the metro timetable.
- Fun: 3 messages were saying that the video message board is really fun.
- Bad: 3 messages said respectively that the kiosk was a waste of money, useless and have bad sound.

- Recommendation: 4 messages were talking about improvements for the keyboard and the today's question app.

Appendix A

User's manual and development guide

In this chapter we are going to give a short description on how to configure the system and how to install it in a PC giving different hints to continue the development of new features for the kiosk.

First of all, given the code of the kiosk, we will need to install the following components:

- Flash Media Server.
- Apache Web Server.
- MySQL Database Server.
- PHP.
PHP PACKAGES INSTALLED :PHP_BZ2, PHP_CUR, PHP_GD2 ,
PHP_GETTEXT, PHP_IMAP, PHP_LDAP, PHP_MBSTRING,
PHP_MYSQL, PHP_MYSQLI, PHP_OPENSSL, PHP_PDO_MYSQL,
PHP_PDO_ODBC, PHP_PDO_SQLITE, PHP_PGSQL, PHP_SOAP,
PHP_SOCKETS,
PHP_SQLITE3, PHP_TIDY, PHP_XMLRPC, PHP_EXIF
- PEAR.
- Flash Builder 4.
- Microsoft Visual Studio C++.

After installing this software you have to check if the installation of the different servers is correct, please check that the following lines were added to the configuration files of:

- php.conf:

```
***** Added by go-pear
include_path=".;C:/PHP/pear"
*****
```

- httpd.conf:

```
#BEGIN PHP INSTALLER EDITS - REMOVE ONLY ON UNIN-
STALL
PHPIniDir "C:/PHP/"
LoadModule php5_module "C:/PHP/php5apache2_2.dll"
#END PHP INSTALLER EDITS - REMOVE ONLY ON UNINSTALL
#
# Controls who can get stuff from this server.
#
Order deny,allow
Deny from all
Allow from localhost
```

- CONFIG PHP & PEAR:

```
go-pear.bat:
@ECHO OFF
set PHP_BIN=php.exe
rem %PHP_BIN% -d output_buffering=0 PEAR/go-pear.phar
php.exe -d phar.require_hash=0 PEAR/go-pear.phar
pause
```

```
PEAR_ENV.reg :
REGEDIT4
```

HKEY_CURRENT_USER/Environment

```
"PHP_PEAR_SYSCONF_DIR"="C://PHP"
"PHP_PEAR_INSTALL_DIR"="C://PHP//pear"
"PHP_PEAR_DOC_DIR"="C://PHP//docs"
"PHP_PEAR_BIN_DIR"="C://PHP"
"PHP_PEAR_DATA_DIR"="C://PHP//data"
"PHP_PEAR_PHP_BIN"="C://PHP//php.exe"
"PHP_PEAR_TEST_DIR"="C://PHP//tests"
```

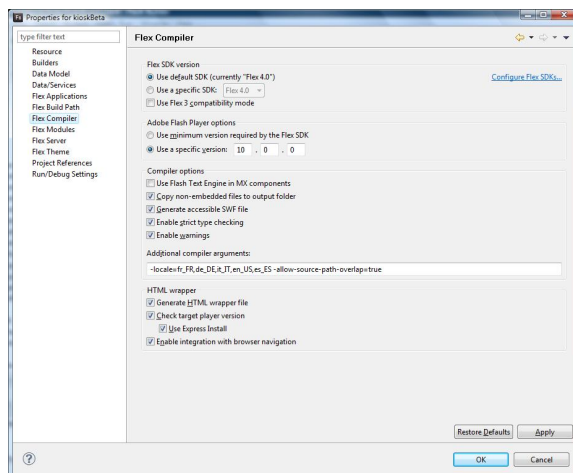


Figure A.1: Flash Builder 4 Flex compiler configuration

With all these listed components installed we can start to configure the system:

First of all, we will copy all the PHP files of the htdocs folder in the same name folder of the Apache file system. We will copy the PHP files of the MySQL database folder into a folder called MySQL where the user wants to place it, after that we can execute these files for creating the database in the Windows Command Prompt: `% php name.php` Once we have the database created and the PHP files into the Apache file system we have to create the following folder into the Flash Media Server file system in order to provide storage for the videos: `E:/Adobe FMS/applications/rlc/streams/_definst_`. Then using Microsoft Visual Studio C++ to compile the RFID reader controller into an exe file and executing the Flex project on Flash Builder 4 or creating a release version will execute the interactive kiosk with the following compiler configuration (Figure A.1).

We had to make some adjustments into Windows XP to have the system running properly. In order to avoid interferences during the execution of the program, we disabled the automatic updates of the antivirus, of Windows itself, of Firefox web browser, and finally and most important of Flash player plug-in.

For getting a proper kiosk interface, we added a plug-in for Firefox in order to get the browser in full-screen at the opening of itself, without bars or menus. We added Firefox to the start menu applications for being executed in the start of Windows. Then the result is, when you start the PC, Windows automatically opens the browser in full-screen mode. Just adding our application as main page of Firefox you get the desired behavior.

A raw command format (PC ->Reader)

SOF (0x01)	Number of bytes	0x00	0x0304	Command + parameters	EOF (0x0000)
------------	-----------------	------	--------	----------------------	--------------

Figure A.2: RFID reader command format

However, there are few minutes during the start-up of Windows that user can see the desktop, so we hid the start bar and all the icons in the desktop in order to avoid mal-intentioned interaction. Although the system is supposed to run without interruption, i.e. without never reaching again this situation of starting the PC, several things like switching off the PC or having a cut in the electricity may put us into this situation.

For managing the different servers into the PC each one has a different port for itself: MySQL server, Flash Media Server (FMS) server and Apache server. The user and password of the FMS is:

FMS user: RLC-KIOSK

FMS pass: RLC

We worked with the RFID reader in the communication model. In this model, a PC communicates with a multiple-protocol RFID reader/writer to read/writer all supported RFID tags. In order to create a connection between PC and a Multiple-protocol RFID reader/writer, the program in PC should setup the following configuration for serial port:

1. Baud rate: 115200 bit/s
2. Data bits: 8
3. Parity: None
4. Stop bits: 1

Next figures (Figure A.2 and A.3) show how the different commands for the RFID reader should be sent in order to configure it and read possible tags.

The communication starts with SOF (0x01). The second byte defines the number of bytes in the frame including SOF. The third byte should be kept at 0x00, fourth byte at 0x03 and the fifth byte at 0x04. The sixth byte is the command code, which is followed by parameters or data. The communication ends with 2 bytes of 0x00.

Finally for parsing the timetable of the metro timetable app we used the following addresses obtained through the QR-code identification in the

Command list:

Command	Parameters	Example
0xFF Initialize Reader	N/A	01 08 00 03 04 FF 00 00
0xFE Get Version	N/A	01 08 00 03 04 FE 0000
0xF1 AM/PM input selection	0x00 – PM input 0xFF – AM input	01 09 00 03 04 F1 00 0000
0xF0 AGC selection	0x00 – AGC enable 0xFF – AGC disable	01 09 00 03 04 F0 FF 0000
0x03 TRF796x enable/disable	0x00 – Reader enable 0xFF – Reader disable	01 09 00 03 04 03 FF 0000
0x0F Direct mode	N/A	01 08 00 03 04 0F 0000
0x10 Write single register	Address, data, address, data...	01 0A 00 03 04 10 15 67 0000
0x11 Write continuous	Address, data, data...	01 0C 00 03 04 11 13 67 46 A4 0000

(a)

0x12 Read single register	Address, address, ...	01 0B 00 03 04 12 01 0A 13 0000
0x13 Read continuous	NR, of bytes to read, start address	01 0A 00 03 04 13 08 03 0000
0x15 Direct command	Direct command code	01 09 00 03 04 15 0F 0000
0x16 Write raw	Data or commands	01 10 00 03 04 16 01 3D 00 00 5A 5B CC 3D 0000
0x18 Request command ISO 15693, Tag-it, 14443B HiBit (ISO 15693)	Flags, command code, data... (as specified in ISO and Tag-it)	01 0B 00 03 04 18 06 20 01 0000
0x14 Inventory	FIFO data	01 0B 00 03 04 14 06 01 00 0000
<i>(EPC)</i>		
0x54 Begin round	No. of slots	01 09 00 03 04 54 03 0000
0x55 Close slot sequence	N/A	01 08 00 03 04 55 0000
<i>(Tag-it)</i>		
0x34 SID poll	Flags command code, mask (as specified in Tag-it)	01 0B 00 03 04 34 00 50 00 0000
<i>(14443A)</i>		
0xA0 REQA	N/A	01 08 00 03 04 A0 0000
0xA2 Select	CID	01 03 00 03 04 A2 11 22 33 44 44 0000
<i>14443B</i>		
0xB0 REQB	N/A	01 08 00 03 04 B0 0000

(b)

Figure A.3: RFID reader command list

metro stops:

```
private var directionRGLF:String=
"http://www.t-l.ch/htr.php?ligne=70&sens=A&arret=EPFL_L";
private var directionLFRG:String=
"http://www.t-l.ch/htr.php?ligne=70&sens=R&arret=EPFL_R";
private var directionRGLF:String=
"http://www.t-l.ch/htr.php?ligne=73&sens=A&arret=EPFL_E";
private var directionLFRG:String=
"http://www.t-l.ch/htr.php?ligne=73&sens=R&arret=EPFL_O";
```

In order to give continuity to this project we are going to show the folder structure of the different components:

The video storage is in E:/Adobe FMS/applications/rlc/streams/_definst_
Other FMS features are in
E:/Adobe FMS/applications/live and E:/Adobe FMS/applications/vod
The database storage is in E:/MySQL Datafiles/ibdata1
The C++ controller of the RFID is in E:/MsTHESIS/SOFT/RFID
The touch screen driver is in E:/MsTHESIS/SOFT/Touchscreen

Looking at the code we have four main folders that we are going to explain:

- `htdocs`: this folder is part of the Apache file system in which are stored all the PHP files used to communicate the Flex app through the web server with the database, the RFID controller or the log files.
- `MySQL`: in this folder are stored all the PHP files necessities to create and delete the database and its related tables.
- `RFID`: the folder containing the C++ code for controlling the RFID reader.
- `kioskBeta`: the main folder of the project containing the Flex application of the kiosk and all its components.

Here is the relation of content of the PHP files on `htdocs`:

- `check_db.php` : file for testing purposes of the database.
- `check_pear.php`: PHP file for testing if PEAR is correctly installed.
- `check_php.php`: file for testing if PHP is correctly installed.
- `deleteMessage.php`: PHP file for deleting a specific message from the database for admin purposes and updating the matching category.
- `flex_comment.php`: PHP file for creating or updating with a new comment of the feedback app, a file containing all the comments of a day.
- `flex_exe.php`: PHP file for calling the RFID controller through PHP and give back the results to Flex.
- `flex_ldap.php`: PHP file for browsing the EPFL directory via LDAP in order to get the associated personal mail account of the given SCIPER number.
- `flex_log.php`: PHP file for updating the `play.log` file and the attribute of played videos on the log table.
- `flex_remove.php`: PHP file for physically deleting the video file of the associated message and sending a warning email to the user.
- `flex_soap.php`: PHP file for connecting to the SOAP Web Service <http://dinfo.epfl.ch/Camipro> and getting the SCIPER number with the id tag of the student card.

- insert.php: PHP file for inserting a new message for the message board app, in case that the person never stored a message, we insert this new person into the database, we update the number of messages of the matching category and the record.log as well as the Log table and the section according to the LDAP retrieved information.
- insertQuestion.php: PHP file for inserting a new question for the today's question app.
- mailsend.php: PHP file for sending mails through PEAR and SMTP.
- query.php: PHP file for querying the video messages by different categories for the message board.
- queryLog.php: PHP file for querying the logs of the message board app in order to show the last five days statistics.
- queryQuestion.php: PHP file for querying the matching question for today's question app, first if there is one already associated for today, if not, search for a not shown yet question and if not, search for an already shown question.
- querySection.php: PHP file for querying all the info of section table for the screensaver.
- queryStart.php : PHP file for querying the last and top video message info for the screensaver.
- simpleQuestion.php : PHP file for querying a question by id.
- test_ldap.php: file for testing purposes, how to discover the structure of the LDAP directory with the university data.
- updateAnswers.php: update the number of answers/contacts that has a video message. Creates or updates the log files. And updates the log table and the section table using LDAP for knowing the matching section.
- updateLog.php: PHP file for updating the number of videos played, answered or recorded in the logs charts or create a new one if is the first of the day.
- updateQuestion.php: for the question being answered in the today's question application update the matching yes, no or don't know attribute and other control attributes.
- updateSection.php : update the number of people belonging to a section/faculty that used the message board in case that the section is already stored, if not create the section and update it to 1.

In the folder named MySQL there is another folder called creation in which we have the necessary files in order to create/delete the database and its tables. The files are the following:

- 0-creation.php: PHP file for creating the RLC_VIDEO database.
- 1-drop.php: PHP file for dropping all the tables of the RLC_VIDEO database.
- 2-tables.php: PHP file for creating all the tables of the RLC_VIDEO database.
- 3-test.php: PHP file for inserting testing initial values for the tables of the RLC_VIDEO database.

In the RFID folder we can find the code for the RFID controller in RFID/C/ISO15693TEST/ISO15693TEST.c, this file contains the RFID control for reading a ISO15693 Tag, protocol followed by the CAMIPRO cards.

ISO15693TEST.c : C++ file for reading a tag under the protocol ISO15693.

In the folder kioskBeta we are going to find the main files of the project and into src the code files divided by folders representing different components.

- 1. bin-debug: debugging files including kioskBeta.html and related swf files.
- 2. bin-release: release files including kioskBeta.html and related swf files.
- 3. html-template: files related to the html template of the project including index.template.html in which you can add or remove the web browser scroll bars for debugging purposes.
- 4. libs: FLVPlayback_2.5.swc external component library for the videos.
- 5. locale: de_DE, en_US, es_ES, fr_FR, it_IT. KIOSK.properties: file of each language organized by a key and a value used by the program as resources for each text.
- 6. media
 - 6.1 assets: photos used along the user interface.
 - 6.2 icons: icons used along the user interface.
 - 6.3 map: epflmap.png

- 7. src
 - 7.1 Components: MapRLC.mxml, VirtualKeyboard.mxml, Web-BrowserLinkRLC.mxml, WebBrowserRLC.mxml, WebcamPanel.as, WebcamRLC.mxml, WebcamViewer.mxml.
 - 7.2 ComponentsAbout: About.mxml
 - 7.3 ComponentsFeed: FeedbackRLC.mxml, VirtualKeyboard.mxml
 - 7.4 ComponentsMenu: ListItemRenderer.mxml, MenuRSS.mxml
 - 7.5 ComponentsMetro: TimeTable.mxml
 - 7.6 ComponentsStart: ApplicationTimer.as, chartsStartScreen.mxml, Countdown.mxml, FLVPlaybackLoop.mxml, ScreenSaver.mxml.
 - 7.7 ComponentsVMB: Answer.mxml, AnswerWindow.mxml, Board-RLC.mxml, FLVPlaybackInFlex.mxml, FLVPlaybackStop.mxml, Identification.mxml, KeyboardShow.mxml, Publish.mxml, VideoList.mxml, VideoObject.as, VideoPlay.mxml, VirtualKeyboard.mxml, WebcamViewer.mxml.
 - 7.8 ComponentsVote: VirtualKeyboard.mxml, VoteBoard.mxml, VoteQuestion.mxml, VoteStats.mxml.
 - 7.9 CustomVMB: CustomEvent.as
 - 7.10 Events: Events.as
 - 7.11 EventsVMB: Events.as, EventsVideo.as
 - 7.12 kioskBeta.mxml
 - 7.13 SkinOverPlay.swf
 - 7.14 SkinUnderAllNoFullNoCaption.swf
- 8. style: bluestyle.css, style.css, whitestyle.css: Application, Button, Panel.

Appendix B

Use study questionnaire and consent form

At the end of the deployment of the system, we conducted an use study among ten people in order to discover the weak points of our design and to be capable of design solutions for these problems. The protocol followed by the users was the following:

Use study protocol

- Free exploration of the kiosk (5 min):
This free exploration show us if the user feels comfortable with the interface and make him get use to the system. This part is probably going to show which parts are not clear to the user and shows the interactiveness with the device.
- Four tasks to be accomplished (5 min - 10 min):
 - Record a video message: In this task we are going to check if it's easy for the user to record a video message, if they follow the steps showed in the screen, as well as if the most complex task can be carried without learning time.
 - Submit a new question in Today's Question app: Low complex task to see if the user can reach other minor applications without problems.
 - Change the language into your mother tongue (if possible) and watch one video of the accommodation section: It was made in order to check the correctness of the different languages and if the user can identify the asked section.

- Write a feedback in the Feedback app: This task try to show us the feeling of the user regarding the keyboard and text input.

- Questionnaire: a simple questionnaire to receive useful feedback.
 - Most useful app:
 - Least useful app:
 - App you like the most:
 - App you like the least:
 - Do you like the design?
 - What do you think about the video quality?
 - Is the kiosk user-friendly?
- Problems encountered: free section for the user to explain the problems that he faced during the study.

Every user participating in this study signed the following consent form:



CRAFT
Informed Consent Form

Please read this consent agreement carefully before agreeing to participate in this experiment.

Purpose of the experiment:

To study how people use the Interactive Kiosk for the Rolex Learning Center.

What you will do in this experiment:

You will be asked to explore the kiosk interface and applications. Afterwards, you will be asked to answer a questionnaire about your experience.

Time required:

The experiment will take approximately 20 min to complete. It will consist of 5 minutes of free exploration of the kiosk, 5-10 minutes for accomplishing 4 tasks, and an additional 5 minutes for completing a questionnaire.

Risks:

There are no anticipated risks associated with participating in this study.

Confidentiality:

Your participation in this experiment will remain confidential, and your identity will not be stored with your data. Your responses will be assigned a code number, and the list connecting your name with this number will be kept in a locked room and will be destroyed once all the data have been collected and analyzed.

Recording:

For the purposes of this experiment and our work in the future, we will be making both audio and video recordings of this experiment. These recordings will be made available to you upon your request. They will not be published without your consent. They will be stored in a locked room.

Contact:

If you have questions about this study, please contact Javier Bello Ruiz, CRAFT, EPFL, email: javier.belloruiz@epfl.ch

Agreement:

The purpose and nature of this research have been sufficiently explained and I agree to participate in this study. I understand that I am free to withdraw at any time.

Non-Disclosure:

By signing this document, you agree NOT to discuss, in any way, the task you are about to work on. Any mention of the material used in this task outside the scope of the experiment may jeopardize the validity of our work.

Signature: _____ Date: _____

Name (print): _____

Additional Consent for Publishing of Recordings:

Please sign below if you allow the researchers who conducted this experiment to **publish images, videos, or audio clips of your participation** in any publication resulting from this work.

Signature: _____ Date: _____

Name (print): _____

Glossary

ActionScript 3.0 Scripting language developed by Adobe used primarily for the development of websites and software targeting the Adobe Flash Player platform, used on Web pages in the form of embedded SWF files, 11, 15, 22, 24, 33

CAMIPRO EPFL personal badge card with a microchip (Standard Legic Advant) RFID (Radio-Frequency IDentification). The antenna that reads the badge is located at the right part of the contactless badge reader. This multifunction system is being installed by the Polyright company member the Kudelski and Securitas group, 7, 10, 12, 14, 19, 22, 37, 72

Cascading Style Sheets (CSS) Style sheet language used to describe the presentation semantics, the look and formatting, of a document written in a markup language, 21

Dropbox Web-based file hosting service operated by Dropbox, Inc. which uses cloud computing to enable users to store and share files and folders with others across the Internet using file synchronization, 32

Flash Media Server (FMS) Proprietary data and media server from Adobe Systems. This server works with the Flash Player runtime to create media driven, multiuser RIAs (Rich Internet Applications), 20, 68, 69

Keyhole Markup Language (KML) XML-based language schema for expressing geographic annotation and visualization on Internet-based, two-dimensional maps and three-dimensional Earth browsers, 5, 10

Lightweight Directory Access Protocol (LDAP) Application protocol for querying and modifying data of directory services implemented in Internet Protocol(IP) networks, 15, 22, 28, 70, 71

localhost Standard hostname given to the address of the loopback network interface, 66

Macromedia eXtensible Markup Language (MXML) XML-based markup language for laying out user interface components. You also use MXML to declaratively define non visual aspects of an application, such as access to data sources on the server and data bindings between user interface components and data sources on the server, 11, 15, 33

MySQL Relational database management system (RDBMS) that runs as a server providing multi-user access to a number of databases, 2, 11, 12, 15, 17, 31, 32, 65, 67, 68, 70, 71

NVIDIA Graphics processing units and chipset technologies for workstations, personal computers, and mobile devices, 17, 18, 33, 34

Projected Capacitive Touch (PCT) Capacitive technology which permits more accurate and flexible operation, by etching the conductive layer in a capacitive touchscreen panel that consists of an insulator such as glass, coated with a transparent conductor, 2, 11, 52

Hypertext Preprocessor (PHP) Widely used, general-purpose scripting language that was originally designed for web development to produce dynamic web pages, 2, 11, 12, 15, 17, 22, 24, 28, 32, 51, 65–67, 70–72

Radio-Frequency IDentification (RFID) Use of an object, typically referred to as an RFID tag, applied to or incorporated into a product, animal, or person for the purpose of identification and tracking using radio waves, 2, 8, 10, 12, 19, 21, 22, 34, 35, 67–70, 72

Rich Internet Application (RIA) Web applications that have many of the characteristics of desktop applications, typically delivered either by way of a site-specific browser, via a browser plug-in, or independently sandboxes or virtual machines, 2, 11, 15

Rolex Learning Center (RLC) Building located on the EPFL campus, the Rolex Learning Center, designed by the internationally renowned Japanese architectural firm SANAA, is at the same time a learning laboratory, a library with 500,000 works and an international cultural center, 8, 10

Really Simple Syndication (RSS) Family of web feed formats used to publish frequently updated works such as blog entries, news headlines, audio, and video in a standardized format, 5, 8, 21

Simple Object Access Protocol (SOAP) Protocol specification for exchanging structured information in the implementation of Web Services in computer networks, 15, 22, 24, 70

Web Service Application programming interfaces (API) or Web APIs that are accessed via Hypertext Transfer Protocol (HTTP) and executed on a remote system hosting the requested services, 22, 28, 70

Web Services Description Language (WSDL) XML-based language that provides a model for describing Web Services, 24

eXtensible Markup Language (XML) Set of rules for encoding documents in machine-readable form, 24, 31

Bibliography

- [1] <http://ch2.php.net/ldap>, visited October 2010.
- [2] <http://ch2.php.net/manual/en/book.mysql.php>, visited October 2010.
- [3] <http://ch2.php.net/manual/en/book.soap.php>, visited October 2010.
- [4] <http://examples.adobe.com/flex3/componentexplorer/explorer.html>, visited October 2010.
- [5] <http://examples.adobe.com/flex3/consulting/styleexplorer/flex3styleexplorer.html>, visited October 2010.
- [6] http://help.adobe.com/en_us/as3/dev/index.html, visited October 2010.
- [7] <http://www.php.net/manual/en/book.exec.php>, visited October 2010.
- [8] Khaled Bachour, Frederic Kaplan, and Pierre Dillenbourg, *An Interactive Table for Supporting Participation Balance in Face-to-Face Collaborative Learning*, IEEE Transactions on Learning Technologies (2010).
- [9] Khaled Bachour, Hamed Seiiid Alavi, Frederic Kaplan, and Pierre Dillenbourg, *Low-Resolution Ambient Awareness Tools for Educational Support*, CHI 2010 Workshop: The Future of HCI And Education, Atlanta, Georgia, USA, April 11, 2010.
- [10] P. Dillenbourg and J. Jermann, *Designing integrative scripts*, Scripting Computer-Supported Collaborative Learning - Cognitive, Computational, and Educational Perspectives, Computer-Supported Collaborative Learning Series, Springer, New York, 2007.
- [11] C. Lewis, *Using the thinking-aloud method in cognitive interface design*, Tech. Report IBM Research Report RC 9265, IBM, Yorktown Heights, NY, 1982.
- [12] Trygve Reenskaug, *THING-MODEL-VIEW-EDITOR*, Tech. Report 5, Xerox PARC, May 1979.

- [13] Peter Pin shan Chen, *The entity-relationship model: Toward a unified view of data*, ACM Transactions on Database Systems **1** (1976).

Apéndice B

Thesis Director's Assessment Letter

EPFL - CRAFT
RLC
Station 20
CH - 1015 Lausanne

Phone : +4121 693 22 75
Fax : +4121 693 60 70
frederic.kaplan@epfl.ch
<http://craft.epfl.ch>



ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Lausanne, le 28 juillet 2010

Assessment letter for Javier Bello Ruiz

I have supervised the work of Javier Bello Ruiz at the CRAFT laboratory at EPFL in 2010. The focus of his research project was the development of an interactive kiosk for the Rolex Learning Center including several interactive multimedia applications (video messaging system, information apps, survey app, etc.)

During this period, Javier Bello Ruiz has shown remarkable technical skills and very promising scientific qualities. His understanding of both technical issues (software and hardware) and scientific challenges has made him one of the key players in our team.

Javier Bello Ruiz is self-motivated, works autonomously and takes good initiatives. In the context of solving difficult technical and scientific issues, he very often suggested interesting novel solutions. Due to his motivation and variety of interests, he got integrated smoothly in an international team involving researchers from rather diverse backgrounds. Moreover, Javier Bello Ruiz is a very friendly and positive person, attentive to divergent opinions.

In summary, during his work at EPFL, Javier Bello Ruiz has shown a great maturity with a very balanced cocktail of initiative, autonomy, creativity, self-confidence and audacity, all qualities that usually takes years to be acquired. In comparison with most of the students I supervised so far, Javier Bello Ruiz's qualities are really outstanding.

Category Rank : Outstanding

Mark : 9/10

Frederic Kaplan, Ph.D-Eng.

Interactive furniture team supervisor.