MiniBands - a collaborative mobile music concept

Fredrik Johansson Petter Hagman

March 11, 2009
Master's Thesis in Computing Science, 2*30 ECTS-credits
Supervisor at CS-UmU: Håkan Gulliksson
Examiner: Per Lindström

Umeå University
Department of Computing Science
SE-901 87 UMEÅ
SWEDEN

Abstract

We have come a long way since industrial society, where specialization and monotone labor was part of a bigger distributed cognitive system. Today society rewards creativity, ideas and versatility. Teaming up with others for additional competence is a natural way to get this process started. One and the same person could come up with ideas, create and test them, and eventually launch them. Demands on music today are high, it needs to be mobile, accessible and (almost) free. The concept of owning music has become a loose concept since the introduction of Napster, Limewire, Kazaa, DC and torrent programs that through different approaches has made music more or less free. Being able to listen to music wherever we are is something that has been taken for granted since the first walkman. But what about creating music? So far this has been a very stationary process that requires expensive equipment and a lot of time. This thesis aims to turn an advanced process like music composing into a mobile concept where beginners, as well as trained musicians, could find their way. We had to research ways to create an application that could reach as many users as possible and be easy enough for beginners, while at the same time challenging for experts.
Music creation is often a collaboration between artists/musicians, each of them an expert (virtuoso) on their instrument. On MiniBands a band still consists of band members, and why should it not? In the concept we propose in this thesis we want to keep some of the individual virtuoso expert approach, but make it easy, mobile and accessible. A band will still consist of different members, each with their own, perhaps favorite, nstrument. MiniBands, the system that we propose in this thesis, allows an easy, mobile interface where users have to develop music in cooperation with other users and eventually publish and distribute it over the web. MiniBands will work closely with artists but even closer with users. Allowing users to try their MiniBands skills with professional musicians will be an incentive that could bring a lot of amateurs to the application. MiniBands will be a completely new way to interact and create music that might be different to that which experts (artists) are used to. This thesis discusses what will make users want to be a part of the Minibands concept and how to create the application for web and mobile. Unlike other popular music games Minibands is not about hitting notes at the right time but about creating actual music. And unlike other musical sequencers Minibands is about creating music together. MiniBands does not intend to replace music-making as it is, but it might give composers an easy way to record their ideas or help to spawn new ones, through an intuitive and playful interface.

Contents

Chapter 1

Introduction

We are two students from the master program in Interaction Technology and Design at Umeå University, Sweden who have written this master thesis in Computer Science. The Interaction Technology and Design program spans over five years and over a wide area of different disciplines which among others include programming, learning about the human factors, problem solving, human computer interaction and studying the future of the web, mobile devices and media as a whole. This master thesis brings together a great deal of these disciplines and also incorporates the experience of working in a real business environment where challenges such as dealing with clients and co-workers with different areas of expertise is apparent at a day-to-day basis. The goal of this master thesis was to develop a conceptual prototype for a new kind of mobile music application. The project was conducted together with Peacock Advertising Agency who originally came up with the idea for the application.
Peacock Advertising Agency (Peacock) is a company with many big clients all around Sweden, who in addition to advertising also develops new digital solutions. An exciting part of working with Peacock was that they had never worked with students before (i.e. we were their first thesis project ever). Initially the project had a partner company, who were supposed to develop the mobile side of the system, the mobile application. This partnership however ended due to restructuring issues and lack of commitment from both parties.
To summarize how the intended system would work it is important to see if from the view of the mobile phone. This is where the limitations, but also the opportunities and capabilities lie. To lay out the idea in short: a mobile application that would make it possible for a group of people to create music together, asynchronously and over distance. The tracks or loops created would finally emerge to a finished piece of music that would be shared to the world through an online-based community. Both the mobile application and the web platform would communicate with the same database and even though they would deal with different service layers, the look and feel of the two interfaces would follow the same design guidelines and also share capabilities and performances and thereby give a consistent feeling from the users' point of view.
On the mobile phone, MiniBands is installed as an application that the user starts when he/she feels like being creative. In the application a user can create an account that contains basic information such as email, username and password. When this is finished the user can easily choose from a number of instruments and compose a track through looping small groups of beats or play the entire track with different beats,

chords and instrumental settings such as pitch and base. When the user feels content with the created piece he/she selects a person from an existing band or someone from the address book and send the track together with a standard invite phrase or a message written by the user. The application connects to the server and uploads data about the track and also creates a message for the intended recipient of the track. When up to five collaborators has laid their tracks the last user select the option to publish the song. The song is thereby made available through the portal directly. The created song can be enjoyed either by choosing to play the song online by the band that created the song or by downloading a media file suited for ring tones or mp3 players. On the portal, top rating and download lists are shown. The users of the MiniBands community will be given the ability to lay vocals or extend and mix their favorite MiniBands tune with an instrument of their choice. Since tunes are created by a number of users this group of users will emerge as the band behind the song. A band can be a one hit wonder that only creates one song or a user can choose to always create music together with hers/his friends. Each user will be depicted through an avatar and visitors can choose to see the avatars perform the songs they have created.
As mentioned above the main focus of the application will be the mobile phone and even though the portal provides a great deal of functionality the center of attention will be the ability to create music from a distance, on the go and whenever people felt like participating. The portal will offer the same opportunities when it comes to creating music, but since the interactive interface of the actual creation process is targeting the mobile phone it is preferable to use just that.
The MiniBands project has three natural parts, the mobile application, the web portal and the server with database management architecture. Our part of the project was to create the web portal where users could come together and share their songs in a new way but we were also assigned the server. This both since we possess the capabilities for doing so, but also since the portal needs the server-side technology to function, even as a prototype. This master thesis was offered to us and we agreed upon doing it since we believed, and still do, in the concept of mobile music collaboration, maybe not to the extent where it would replace location based music collaboration, but definitely as a complement that some day will make music a process that can be created whenever creativity strikes.

1.1 Goal

We had a couple of different goals going in to this project. More precisely one can say that smaller goals were identified during the process of this project. The primary goal was to define functions and find a suitable environment to develop the application.
One goal was to find out the criteria for user interaction (and collaboration) that our software would promote. The goal includes finding out what motivates users to collaborate and how the interaction should work and be promoted to be optimal, based on given limitations, such as the fact that the application will be implemented on a mobile phone. This goal was reached by an in-depth study of recent research in the mobile music creation area, and also by basic studies as to what drives us to collaborate and what the important elements in such a situation are. This research also came to follow how music applications is developed today and what seems to be done right and wrong.
Another goal we set out was to create a system that could be used on different platforms, both mobile and stationary. We wanted to explore the possibilities to create a software that easily could be translated (or perhaps need not be translated at all) to

work on another platform. The system should be a three layered solution. What general design proposals exist for such an application? What is used today and where is the future heading? Should the web interface be so simple that it's able to function both on regular applications and mobile browsing, or should there be two different solutions? Money is generated when people use the mobile-interface, through data transmissions over the mobile net, how do we promote the use of it in our design? Should we? Who are the main stakeholders/users of our design? Is it the end user or is it the company that will make profit through mobile Internet subscriptions. These were all questions we had to ask ourselves in the beginning of the process. There were also questions on what technologies to use. When developing a system that will be used by different platforms and devices, what programming language is preferable? What different development environments are available? Which one will suit our solution best? How will different languages in different layers work together optimally?

1.2 Structure

Below follows a description of the report's disposition

Chapter 2 - Background

Gives a description of what technologies and services that are available today. It discusses the recent changes in the music industry. Deeper analyses are provided on mobile music collaboration and web applications for mobiles.

Chapter 3 - Design process

Describes the steps of the project from idea to prototype.

Chapter 4 - Results

Presents the end results, the prototype and what goals we reached. Concludes the project.

Chapter 5 - Summary / Reflections

Discusses the project, what roles we were given and whether it was a success or not.

Chapter 2

Background

To be able to start development on the application, it was necessary to research the users that the service is targeting; mobility, usability, collaboration incentives and what already exists. We wanted to create an application that fulfills a new need and does not steal focus from other music applications today. So, in order for developent to start these fields had to be researched further. Another reason for the background research was to create guidelines for future development. Since our task was to create the web portal, we needed to study some techniques that could help the development of a mobile application in the future. This section of the thesis report covers the individual theoretical parts that are mandatory in a thesus for the interaction and design program.

2.1 Intro

The average Joe of today has tons of friends, online and in the real world. Some are fleeting connections that merely constitute a "friend"-connotation on some web community whilst some are lifelong connections with family or close friends. Today there are a lot of different web communities,applications and technologies that helps us keep track of it all, these communities however also facilitates a lot more than just this. Some are trying the versatile approach, in an attempt to cover all sorts of interactions between users, e.g. facebook that has its own platform for developers to make their own applications, also the graphic user interface is very free to create for users to personalize their own presentation. Other communities are more specialized to one task, e.g. myspace, which mainly targets the spreading of music and music videos. The already existing web communities basically have the same approach, a rich user interface with many interaction possibilities. However, viewing these communities through the browser of a mobile phone is more complex, simply because the interaction and graphic presentation possibilities are limited, i.e small screen, limited number of buttons and no analog input, at least for the most popular devices today. Music is in high demand on the web, this includes pure downloads and streamed music like earlier mentioned myspace.
Other music communities use connotations or tags to classify music and record users music choices to generate a personal music profile e.g. last.fm. The mobile interfaces on these communities are non-existent or poor. A pattern emerges from the popular use of electronic devices. Friends and music seems to be of key importance in this pattern.
Today mobile phone users are flooded with new applications and ringtones, just consider jamba or appstore where users pay to customize and personalize their own handheld de-

vice, at the same time computer users are participating in online communities to keep track of their friends (real-life friends or just internet related friends).
Mobility is a very important aspect of music today. The fact that mp3-players are incorporated in (almost) every digital device makes one realize how far mobile music has gone. Even though mobile music devices are used by a great deal of people every user's music profile is different. Making your own collection of music based on what music you listen to and like is a trend that we see a lot in web applications, like pandora. These web-based applications are however created to target the stationary web and some of them are not intended to be mobile at all. Last.fm and Spotify are however two services that are attempting to become mobile. Last.fm does this by listening to (scrobbling) the music played on the user mobile device and sends the track to the main server to keep track of the favorite songs and adds them to the users profile. Spotify has not yet launched its mobile application but it will work in a similar way as the stationary version, it will be able to connect to the database where the user can search and compose playlists and play music. Another approach that some applications are using today is streaming music from the users own PC. This approach doesn't involve logging in to any site, buying songs or getting commercial breaks in between songs. DOT.TUNES, SimplifyMedia and Orb are a few of the applications that deliver this service. DOT.TUNES and SimplifyMedia are only compatible with Ipod Touch and Iphone while Orb has a broader spectrum of compatible devices. Orb isn't limited only to music, it also allows for access to photos, videos, live television, and other digital content from the home PC at anytime and more generally from any internet-connected device, such as a mobile phone, PDA or laptop.
In these times companies are easily left behind when trying to limit the use of distributed music (or other forms of media) to one location or to only allow access from "clumsy" technology (cd-players, laptops, desktops etc.). This is a thing of the past and something that customers are beginning to complain about and directly choose not to conform with. The services mentioned above are trying to bridge these factors.
When discussing mobile music, there is one area where the development hasn't gone far enough. The fact is that even though things are getting better when it comes to consuming, sharing and enjoying music, the creation of music is still very localized and in some ways limited. On a mobile phone services are limited to the DJ programs where they have a few sounds and a static timeline on which the user can distribute these sounds. Often these applications lack a way to collaborate, share or reuse the created material. The idea is that when the time comes to launch MiniBands on the market it will revolutionize the way we create and share music. If the penetration of the MiniBands concept gets large enough it is possible that certain tracks can be sold to other platforms or media, which can make creating music through the MiniBands system a viable income for music creators.

2.2 Problem specification

To specify the intended result of this project, we first have to structure the different parts of the thesis work. On one hand, the intended result of the project is to create a web based prototype of the system that can be used during promotion and demonstration of the concept to venture capitalists.
On the other hand, article studies had to be done to get enough background information to start development of a full application with accessible interfaces both to mobile and stationary computers.

First we wanted to know what technologies are needed to make the system mobile and second on ways to entice users to collaborate to create music. An application of this magnitude requires a system in three layers: an interface layer on top, a communication layer in the middle and a database layer in the bottom.
To create a three-layered system that not only allows users to communicate with each other but also can be used to create and publish music together with others is another goal. Mobility and user-friendly interaction techniques are vital for an application to survive on today's market that is flooded with applications (since everyone can be a software developer - compare Apple Appstore). The application would also have to be good enough for people to actually feel that it can be a part of real music creation and viewed as an extension to current collaborate music creation technologies. A service that can be extended and built to generate income through selling songs, extensions and rights. A study has been performed that will show what will motivate users to collaborate and what makes them see the application as an instrument rather than a game (see section on Mobile music collaboration).
Another goal with the application was a good market penetration. An article study was performed on the techniques used to run (render) web-applications on both mobile and stationary platforms. The target for this study was to study transcoding. Transcoding is a way to adapt code to fit other devices. And, to successfully adapt something to a specific platform, with specific properties, an identification of the device has to initiate the adaptation process. The usability goals for the application were simply to create an easy navigation through the different application states and interconnectivity in a way that the user never gets stuck. Furthermore the application should be designed according to standard practices for portals to promote user recognition. This supports the phenomenon of "recognition over recall". Our ability to recall information is less efficient than our ability to recognize it from some visual cues.
Other, less prioritized goals of the project was to actually sell the concept. and to start user tests on the application for further development of Minibands.
In short the goals were

- create a prototype
- research mobility technology for market penetration
- study what motivates collaboration
- easy navigation through the application for usability
- find venture capitalists to invest in a full size application

2.3 Music industry

2.3.1 Music consumption is becoming mobile

The music industry has changed over the last couple of years. Since the digitalization of music, media focus has been set on mobility and accessibility. The future audience no longer waits to buy a record in a music store, they hear about it, pre-listen to it on the internet, through their computer or handheld device, download it, share it and wear it (portable players). Many use illegal techniques to get music for free and as a result the music industry suffers economically. During a period of time it has been looking dark for music artists since they no longer can make as much money as before

by only being good in the studio and producing records. One could argue that this is a good thing, since real artists who loves spreading their music through performing live still can make their living on music while super produced untalented studio musicians suffer, perhaps will we see more genuine musicians? One could also go to the other extreme in the argument and say that every illegal download is a theft and should be tracked and punished. There are two different camps in this issue, those who believe data and information should belong to no one and those who believe data theft is a crime. The music industry is however beginning to accept the fact that people will not stop using Internet to feed their hunger for music and go back to consuming their music in a music store. People will get their hands on the music they like, whether it is through downloading the music illegally or legally downloading the music using other distribution channels.
The way we access music has changed. No lawsuit is going to change that. This lawsuit is a pathetic move by an industry without any creativity, as it fails to adapt to a new reality. - Erin Harrington, Fairbanks, Alaska, April 23, 2003.
The main factor to consider now is simplicity and availability. Who will be able to give the customers what they want, in the fastest manner, with least amount of hassle and with a feeling of confidence and quality. [21]

The industry is catching up by providing models where song files can be copied and re-sold legally. So-called "superdistribution" is designed to allow music fans to share songs temporarily with friends. Users will be given the opportunity to pay for the music if they want to keep listening, giving both the original copyright holder and the referrer a cut of the profits into the bargain. [33]

A few techniques and applications today show us that by making things as simple as can be, money can still be made through selling music. The important factor is giving the user the opportunity, at the right time. One application that plays on the human characteristic of impulsiveness and instant gratification is TrackId that can be found on many of Sony Ericsson's latest mobile phones. The application can in a seemingly magical way, in reality it connects to a enormous database called Gracenote, tell the users which song that is currently being played only by listening to it for a few seconds. This can take place in a fully packed bar, a concert, in an elevator, or in front of the television set. When the application gets a hit on the song played, the users gets the name of the artist and the song displayed [12]. The next step yields even more datatransfer, which means even more revenue for the operator. With each successful search hit on a song the user is posed with the question: Do you wish to buy this track? Here we have a clear case of impulsiveness and furthermore instant gratification if the user chooses to accept the offer. In short an ingenious application that everyone from users to companies loves.
Today there are even legal ways to access music without buying an album, it might cost a little bit of money, or a daily dose of targeted commercials (Spotify, free version) but if you want to support musicians or just don't like risking getting fined for digital theft there are a couple of applications and websites like Spotify that offers just that. Last.fm and Pandora rose to the music sky a couple of years ago and for people living in Europe Last.fm is still accessible and gets more advanced and commercially acceptable by time. Through the application interface the user can't choose the exact songs that will be played but rather which style or type of artist they feel like listening to. The last.fm platform keeps tabs on all music the user listens to, not only through the application itself, but through other media players on the computer or even on a mobile phone or

other hand held device. The user accepts this process of collecting information, called scrobbling. The result is creating a musical profile of the user. When enough songs have been scrobbled the system has a pretty good guess what the user likes. Tracks recommended by neighbors and events are generated along with a list of all tracks played and a playlist called the users library, containing tracks that the user has listened to, that can be played by other Last.fm users or by the user [23].
Event though these web based applications, and other ones similar, probably has a great future ahead of them Last.fm and Spotify has one limitation and that is the lack of a mobile penetration. With the coming release of such applications these platforms are becoming serious contenders to locally stored music such as mp3-files on handheld devices. Spotify requires some disk space, but in comparison to the resources on mp3-players of today that is irrelevant[24].

2.3.2 Creating music is becoming distributed, decentralized

As will be discussed in the section on Mobile Music Collaboration, music is a collaborative form of art and can be created in a number of different ways. It could be spun from the gathering of musicians and later evolved into a song, it could be the work of one talented individual that writes the entire piece in solitude, but perhaps performs the piece together with a band. Both these scenarios will almost certainly always exist. To say that creating music is becoming distributed is to say that music is being decentralized with parts spread all over. For instance, a band does not need to jam together anymore, at least the members do not have to be in the same physical space, they can be located anywhere on the planet as long as they stay connected. This perhaps applies most to digitally created music but since all music today finally will end up as digitalized the phenomenon is spreading. Along with the rise of computer communication that gave birth to the somewhat structured organism that is the Internet a new form of human communication was born. Nowadays when people are trying to contact someone that they see is online the easiest way is to send an instant message, or if the information is too large or complex, an email. With Internet, people can connect in a new way and also stand out in a way never possible before. For musicians this can be a lifesaver. Perhaps their particular form of music they create is hard to sell in their own country and how do you make a name in a country you never visited before. A new way of spreading music is by getting the word out on the web. This can open doors and create a fan base that before only was possible with the right connections. With enough money a well-built webpage and collaboration with a super producer in another country can make a huge success. Today producers seem to be more popular than artists, or at least the producers are looked upon as having the recipe for hit music. With enough money an artist can buy a beat from a super producer and expect a success. The artist and the producer never even have to meet each other. They send samples of their music and collaborate online. A less business-oriented way of reaching the crowds can be seen on MySpace where musicians have their pages where fans can enjoy free songs but where collaborations between musicians also can start.

2.4 Mobile Music Collaboration

Since the MiniBands concept originates from an idea of a mobile music application where people had to collaborate in order to create a final piece of music, a deeper study of mobility, collaboration and music creation was performed. This was done to create a

solid background to form an idea on how the intended system would be implemented based on important aspects, found design guidelines and specific device abilities.

We believe that musical creation that will happen asynchronously in a process where collaborators are situated in different locations.

Today we see it everywhere; on the way to work, on the bus or perhaps when working out we see people listening to music. They are carrying along their choice of music, played through their headphones. This may not be so different to how it was 10 or 15 years ago. The difference is rather how the music is stored, and where. Today most of the music is digitalized and stored on a mobile phone, a mobile mp3-player or even streamed through a centralized web based server. Since most mobile phones and mp3-players have the ability to send and receive files by using different communication technologies it is easy to share and download existing pieces of music.

We are beginning to see that all of our portable devices are coming into one universal gadget with capabilities far greater than ever expected. This gadget is not really a new one; it is the mobile phone. It looks similar to what it did 10 years ago but today we see our mobile phones in a new way, as an important piece of technology that is used in many ways and one that we rely upon [34]. It is looked upon as a lifesaver, a way to keep in contact with friends and family and perhaps even a way to avoid face-to-face communication. The mobile phone has made our lives easier in several ways but some things cannot be performed without real human contact, or can they? Humans can collaborate in various ways and with different outcomes and reasons. Some collaborative processes are performed in order to reach a common goal, some others just happen when people are trying to reach a personal goal that happens to emerge in a way that can be described as collaboration.

This thesis discusses the creation of a mobile music application and therefore we will now look further into music creation as a collaborative creation process and find out how humans collaborate on a distance. What is needed is both a technological and psychological point of view on how issues can be bridged in order to create music through the mobile phone, on the go.

2.4.1 The mobile human

Humans today tend to rely on their mobile phones. We rely on that we will have the ability to communicate with others whenever we want to. This means not only relying on the mobile phone, but also that others will answer when we try to contact them, almost like if they were in the same room. Since most people live in areas where the mobile networks are highly developed we get accustomed to having a good network connection and we feel that we can travel freely and still keep in touch with family, co-workers and various other human connections. With the spreading of 3G the Internet is reachable to people almost wherever they are, with the ability to cater and support for most of the virtual needs of a person on the go.

Often the bit rate of 3G is enough and when it is not sufficient WIFI hotspots can solve the problem. However, the fact is that these hotspots have not reached a high enough level of concentration to support WIFI access outside of cities or even outside of certain hotspots inside a city makes it an unreliable alternative to chose as primary connection channel. A mobile phone with 3G and a flat network subscription is enough to make the Internet access from the mobile phone ubiquitous, where the connection to the web is required and anticipated to work whenever, without setting up the connection, choosing a suitable connection alternative or entering user names and passwords. This

makes it possible to implement applications and services that connect to the web when needed and therefore can let the user use web based services through the mobile phone.

The mobile phone is becoming an extension of us, a way to define who we are, at least on a virtual level. We define our mobile persona through the way we write our text messages, which ring tones we use, which mobile phone we are carrying with us or perhaps what designer case we choose to put it in. With the release of smart phones with touch interfaces people can for the first time interact in an intuitive way and with the Apple iPhone people can even in an easy way customize the mobile phone itself by adding and removing applications. By this they are customizing the mobile phone itself.

Today one can say that we are mobile cyborgs, human bodies extended by an impressive piece of technology, that whenever faced with a task that before required one to be stationary we now try to solve by embodying our mobile phones with yet another ability, and by this giving ourselves this ability. We are heading to a mobile state of being.

2.4.2 The collaborating human

Humans are said to be social animals. We seek contact with other humans, and we are used to cooperate to achieve our goals. By interacting in various ways, we can share information and exchange ideas and opinions. Both in professional and private settings, we use communication as a mean for cooperation. Cooperation is vital for productivity at workplaces, school or inside the family.

Defining collaboration

When working with music, can one say that a band is cooperating, or is the proper term instead collaborating? What is the difference between cooperation and collaboration? It is important to separate the concepts:

Communication refers to how people understand each other and how information (not just facts, but policies, prospects, rumors, feelings, failures, and all other human experiences) is transferred. While lack of communication tops the problem list in most organizations, the diagnosis is a facile one for many reasons [1].

Coordination, like communication, begins with an assumption of differences, as in different persons working together. As in athletics, we are coordinated when the arms and legs move together. Everything falls into balance if not symmetry. Coordination is about efficiency. Unlike communication, however, coordination looks to inform each unit or part of the whole as to how and when it must act. In many cases, coordination boils down to two conditions: that people and units know what they are to do and when they are to do [1].

Cooperation is important but so is divergence. If someone has a very different idea on how to contribute to the group is then this objection non-cooperative to raise? Much of creativity comes from the sparks of disagreement, dissent, and even conflict. Cooperation too often becomes a call for increased socialization to a culture, not a prompt for high performance. Also, one opposite of cooperative is competitive. Consider that virtually all of what we call strategy is about gaining competitive or comparative advantage. Cooperative thinking is rarely the same as strategic insight [1].

Collaboration is something similar but nevertheless separated from each of the C words profiled above. Unlike communication, it is not about exchanging information. It is about using information to create something new. Unlike coordination, collabo-

ration seeks divergent insight and spontaneity, not structural harmony. And, unlike cooperation, collaboration thrives on differences and requires the sparks of dissent [8].

The general thought definition of collaboration is according to Wikipedia *"A recursive process where two or more people or organizations work together toward an intersection of common goals"*. There is however more to it that just that, collaboration is not only about agreement. It is about creation. Michael Schrage writes in his book, Shared Mind - The New Technologies of Collaboration that *"collaboration is the process of shared creation: two or more individuals with complementary skills interacting to create a shared understanding that none had previously possessed or could have come to on their own. Collaboration creates a shared meaning about a process, a product, or an event. In this sense, there is nothing routine about it. Something is there that wasnt there before."[30]*

One can say that innovation is about divergent thinking and the creation of something new, and collaboration is an essential tool for achieving it. Collaboration is not about large numbers of people who have widespread input or come to consensus. It is about a small number of persons who bring distinctive, if not unique value, to the creative process.

Extending collaboration through technology

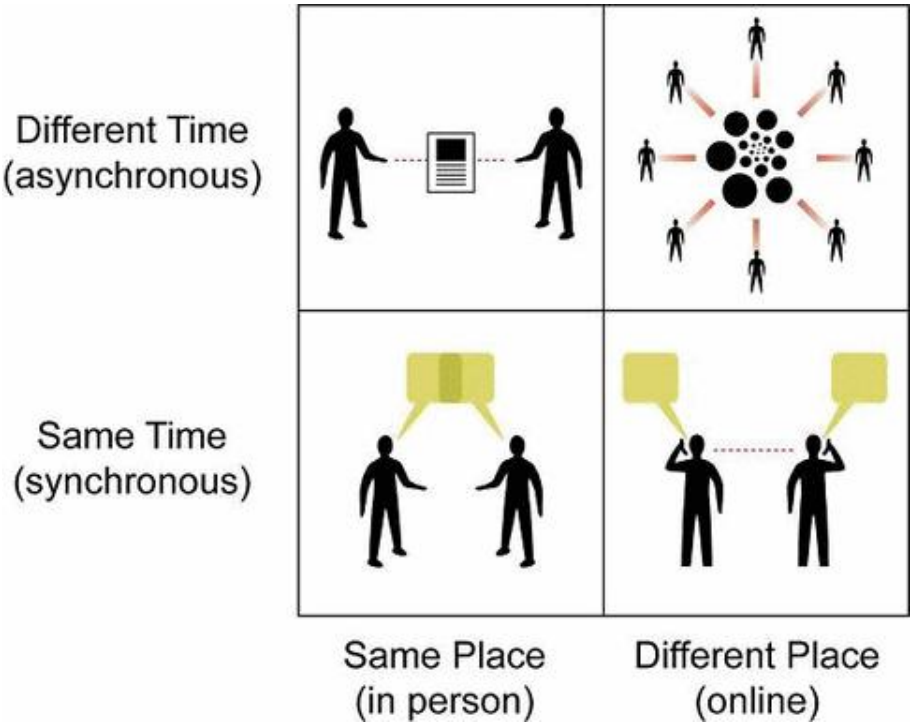

Figure 2.1: Time Place Matrix

The definition of human collaboration is as mentioned above not about exchanging information, but about using information to create something new. Collaboration does however not look the same in all situations. It is heavily depending on the task at hand,

if collaborators are located on the same place at the same time, and how collaboration is supported, see figure 2.1.

Many research projects have investigated in how collaboration can be improved using technology [2]. Within this research the findings often say that the ultimate collaborative environment is were all participating parties are situated close to each other with the ability to freely use their own voice and body to transfer their intended message of information. Of course technology can make a great difference even when people are situated together. Examples of this is a large touch screen that enables everyone to interact, all at the same time, or a system that is listening in on all communication to make it possible to go back in time to bring back decisions or ideas that were lost somewhere in the process.

Today it seems that our society increasingly is relying on geographically distributed collaborations for human interactions. Whether it is business, science, or different kind of creative art. These collaborations improve communication among individuals with a common purpose, the promote:

- sharing,
- spontaneous collaborative efforts,
- development support,
- distribution of information.

The Internet supports distributed collaboration teams in which collaborators interact visually and verbally, augmented by additional tools and services such as virtual reality and immersive environments. Distributed collaborations increasingly require interactions to be as natural and realistic as possible

There are situations where technology is needed in order to make collaboration work. This can be collaboration between co-workers that are located on different places, or perhaps accessing a shared space at different times. In such a mediated setting, the whole idea of working together becomes fragmented to a series of recorded messages where the fine print and visual cues only present in face-to-face communication is lost. Technology-mediated settings bring the concept of virtuality to the repertoire of collaboration, but the fact is that the human-human interface remains centre for any kind of human collaboration. It is in no way possible to imitate the richness of natural human interaction and communication in a technology-mediated way, because part of the sensory information requires immediate presence. Nuances of smell, taste and personal touch remain local, close to the body. Perhaps new ways of sending such subtle cues can be invented, ways to extends how non face-to-face communication is nuanced. Emoticons is one attempt at this. [35]

A technology-mediated forum for collaboration is a kind of representational augmented personal space. Therefore the life situation of the person is the ultimate platform of any groupware and communications technology. The person is the principal site of personal coordination of communication, collaboration and identity [20].

Mobile Collaboration

Simple forms of mobile collaboration take place all of the time in our everyday lives. Communicating through our mobile phones by speech, text, images and other various media are all ways to send, receive or just distribute information. Often we do this in

order to synchronize our ideas, feelings, states of being or information needs. These informative actions are however not always a part of an out-spoken collaborative process. One can of course discuss whether or not all human communication is a kind of collaboration, but in this thesis collaboration is considered a defined process existing during a limited amount of time and where the intended goal is to create something new.

With the growing popularity of mobile technologies and the increasing use of groups within organizations the question remains as to whether mobile collaboration technologies provide benefits over face-to-face and conventional computer mediated technologies to such collaborations. One of the main benefits is clear, that is the almost constant ability for communication and sharing of information. Even thought it feels that the benefits would be easy to put into word the disadvantages are often easier to formulate. A couple of disadvantages compared to face-to-face collaboration are lost visual cues and the lack of an information flow without interruptions caused by network errors or things happening in the mobile context.

When developing mobile applications intended to support collaborative activities there are a number of different ways that people can collaborate. In the system presented in this thesis people will be collaborate over a distance and asynchronously. The concept by itself bridges one of the largest issues when it comes to mobile collaboration, namely synchronous communication where turn-taking and network problems are common issues. A question that needs to be addressed is what media types that will be used when users are communicating between each other. There are some media technologies available on most mobile phones to choose from:

- Video: A rich media form, but has problems with gaze, gesture and non-verbal communication.
- Audio: Conveys meaning well, but not necessarily location.
- Text: Good for synchronous or asynchronous communication.

The MiniBands concept of course has to use audio to communicate the created music between users, but the communication between the collaborators inside the application is preferably done through text. Before this fact was discovered the idea was that communication between users should be done through SMS, which fits well with the type of collaborative work that will be performed. Whether this was decided because of the fact that it was the only communication alternative for the majority of mobile phones or not, it actually agrees with the overall concept of asynchronous communication. One of the major reasons why the Short Message Service (SMS) has become so widely popular is the fact that it is an asynchronous way of communicating. When the SMS has been delivered the act of transferring the intended information is over. The sender knows that the message has been received, but not when or where it will be read and when a response can be expected. This might seem like negative aspects, but is actually looked upon as communicative advantages in a collaboration process where participators is collaborating in different locations and at different times.

This all supposes the fact that the user wants to use the application. There are however important to find the potential incentive for the user. There are a number of key abilities that the application needs to accommodate [2].

- Promote spontaneous collaboration: In order to make the application truly mobile most if its functionality should be able to be used whenever the user feels creative.

Limitations such as a minimum bandwidth should be limited in order to promote spontaneity.

– Enable trust: The user should at all times feel that personal information is secure and that the application does not act by it own, but rather ask for instructions.

These are both important aspects to consider, but do however not create an urge of creative expressionism. In the MiniBands concept three Unique Selling Propositions (USP) have been identified and need to be implemented. These are:

1. An application never to grow tired of:

   There is no final level in Minibands. No end boss after which you do not want to play Minibands no more. It is a game concept that grows with its user, and thus provides him/her with endless possibilities to evolve and create new and innovative music.

2. An application where the user can make a buck or two:

   MiniBands is a game that gives its users the possibility to make money. By producing songs that other people download the composer will earn money. Also, by making instruments for Minibands that other people download the user will make money. And, with the possibility to make money comes the incitement to improve and involve others.

3. An application that you can not play alone:

   Minibands is a game that brings out the essence of why mobile phones have been successful because of relationship building and communication. To be able to play Minibands you have to communicate with other people, and what better way to do it than on a mobile phone.

Mobile multi-tasking

One can say that we see our mobile phone as a multi purpose device, which means that we will easily adapt to the idea of performing more tasks with it. Some tasks such as seeing what food is currently in the refrigerator may not be available to perform while mobile. If such an enabler would be created one most certainly would presume that people would find it useful. It has however been questioned whether some actions belong in certain contexts or is even wanted there.

Since music is already being consumed at a large scale through mobile phones the phone is seen as a music device that actually can play back music so well that it is safe to say that it is beginning to replace the single use MP3-player[10]. This would make it a good medium in which to also incorporate the task of mobile music creation. There are already a number of creative music applications that are highly popular today. Most of these applications target a single user performing multiple instruments. Even though this goes against MiniBands collaborative notion of music creation, the main difference is really that each user only can create one track and therefore have to join forces with someone else to create a piece of music.

2.4.3 Music as a collaborative art form

Music is a social art by nature. Whether in songwriting, recording, or performing, collaborative acts are at the heart of the musical creative process, independent of technology [32]. Even though music as an art form is independent of technology, technology can help. Collaborative creation and the production of open and continuously evolving works are two of the most appealing artistic breakthroughs the Internet can offer to music composers and creators in general. The ability so share work and to re-create old compositions makes Internet a gold mine for musical artists.

Humans value music in a special way. It is considered to be something similar to or equal to art. A way to express the very being of a human. Reimer [28] says that there are five dimensions of musical value, namely:

1. Music is ends and means. It is valuable in and of itself.
2. Music encompasses mind, body, and feeling.
3. Music is universal, cultural and individual.
4. Music is product and process.
5. Music is pleasurable and profound.

Often today, in a world where consumption is king, music is seen as a product. Only a few people gets the chance to learn an instrument, but most people listens to music or has an opinion on how music should sound. The MiniBands concept is defined as a mobile music creation process available to everyone with a mobile phone. Everyone should be able to create music easily and perhaps have the joy to expand their value of music further than as a product.

Nowadays when music is mainly produced or re-mastered in a digital way, artists and producers can work together without even meeting each other. It often seems that the fastest way to success in the music business is to hire a famous producer that makes the beats. These beats are later incorporated with the voice of a singer or the sound of an entire band. This can be done over a distance and often is. By this example one can point out that even though music is a collaborative art form it could be created by a number of people over a distance.

Are creative experiences collaborative or are they performed in solitude? Many of us believe music to be a collaborative creation process where an entire band sits together and comes up with ideas and test different approaches to find the right path to their masterpiece. The fact is that there are often a period of time when the musicians sit alone and work with their own part with their instrument of choice before rejoining with the band to stitch it all together [22]. It is that belief that the MiniBands concept originates from.

Mobile music technology

The MiniBands concept is mainly focused on creating an applications for existing devices such as mobile phones and smart phones. Other mobile music devices such as the Pacemaker Pocket DJ [1] is also a great example of how digital music can be created, however, within this thesis such a device is not considered a viable solution to how mobile digital music can be created.

The idea for the MiniBands concept was as mentioned above an application where users can create music together, asynchronously. Most creative mobile music application of today do not support the fact that real life music often is created by a number of collaborators, each with their own special ability to control an instrument, a virtuoso. There are a few examples though and some of them not only support the task of creating music while mobile, but also turns the state of being mobile to a part of the creation process. SonicCity [11] is one approach where users creates music simply by walking through an urban landscape.

When creating an application that will simulate the creation of real music, how important is the step where the music notation is created and written down? One can see that music notation in music applications creates an unfortunate barrier to participation for novice users. And, when children are learning an instrument, many choose to quit when the music notation part begins [14]. This is the point where curiosity can be lost and where only the truly dedicated stays. What level of user knowledge in music notation should the application target? Of course an application that attract both beginners and real musicians would be the ultimate choice. Perhaps there could be a way to switch between novice and advanced mode. Regardless of what route is chosen the application should not demand that the user knows how to write notations but instead replace the music notations with an informal GUI. An interface that feels more intuitive and can provide instant feedback to the user. An example of such an interface is Hyperscore [22]. This concept invites users, mainly children, to explore music in a more intuitive way. The concept of an informal GUI should be taken into account when developing the mobile application, perhaps not with a result as non-instrumental as Hyperscore [22] or as informal as SonicCity [11], where the interface is the context, but with enough intuitive visual cues presented on a screen to entice both amateurs as well as people familiar with music creation.

Choosing an informal GUI

Does an informal GUI equals a intuitive interface? This of course depends on the specific implementation, but one can certainly say that this is not always the case. To be able to create an intuitive interface the device hosting the application must be equipped with the necessary features. The MiniBands concept is however supposed to target the average mobile phone user of today and the average mobile phone of today does not support interaction methods beyond the ordinary keypad, but what about tomorrow? If we look at the high-end devices of today we might see how the average device of tomorrow will look like. Maybe this is the way to go instead of limiting the application to yesterdays standards? The MiniBands concept does however have some limitations regarding this. Namely that the web based application should be able to imitate the music creation process on the mobile phone. This makes interaction methods such as the usage of accelerometer data and various other sensors, that can be considered as mobile specific, a poor choice as a main interaction method. However, the application interface could be designed so that it could extended by such sensor data. For instance, instead of pressing key 1 on the keypad, the mapped visualization of key 1 on the screen can be touched on a touch screen devices. If the device has an accelerometer the user can move the device as a drumstick, to create the same action as pressing key 1.

Determining application type

When researching current mobile music technology there is a vast collection of available applications to choose from, and a large quantity of these use an informal GUI implementation without musical notation. When looking at a mobile music application it is important to establish where the limitations come from, the application or the device hosting the application. For instance, if an application is running on a mobile phone without touch screen, then an intuitive touch interface is impossible to implement. To be able to find a suitable interaction methods for todays mobile phones, with their possibilities and limitations it first has to be established how users should work with the application. There are a number of categories among the available applications of today. These categorizations are:

- Phone Keyboard Applications (CAT #1) These classic applications enable a user to play an instrument by pressing on the number pad which is mapped to specific notes on an instrument. The model severely limits the creative possibilities because of the inherent limitation of the keyboard size and the small number of keys [7].

- DJ Mixer Applications (CAT #2) This type if application allows non-muscicians to mix pre-recorded and user recorded tracks in a form of a DJ mixing console. Using a simple interface, users select prerecorded music elements and mix them together for single or multi track playback. The application is simple enough for any phone user to enjoy [7].

- Audio Sampling and Synthesis Applications (CAT #3) This class of application allows non-musicians to generate music by either sampling sound from the world using the mobile as an instrument or by artificially generating sounds using audio synthesis functions [7].

- Studio Sequencer Applications (CAT #4) These are complex applications that use the music sequencer model in which a sophisticated multi track interface enables users to record and play back sequences. Additional functions enable mixing, looping and special effects. These applications are too difficult for the casual user [7].

All these application-types use the standard mobile phone interface, namely keypad and on screen buttons only. When looking at the details explaining each of these application categories it is evident that the MiniBands concept will utilize a mix of features from these categories in order to perform well. The main difference between the categories are in the way the user is intended to interact with the application and what the level of musical skills the intended user is expected to have. When looking at the pre-defined concept of MiniBands the application category with the best match it is the Studio Sequencer Application (CAT #4) and the Phone Keyboard application (CAT #1). In most applications of CAT #4 it is said that the user can record and play back loops. These loops would later be stitched together to one piece of music. These applications are however considered to be too difficult for casual users [9] and the fact that the interaction methods used in these applications are not suitable for the MiniBands project make CAT #4 a poor choice to follow completely. This is where CAT #1 comes in. This kind of interaction enables a user to play an instrument by pressing on the number pad, which is mapped to a specific note or sample.

2.4.4 Summarizing the factors

To summarize how the mobile application should be designed, both to fit the MiniBands concept and the numerous factors when considering music creation and human collaborations mentioned above, a short summary will be stated below:

- Music on the go

  We have for many year carried our favorite music with us . Ever since the release of the Sony Walkman this has been a fact. Nowadays we have a far more powerful device in our mobile phone that has taken on most of our daily tasks. It has even done such a good job with the music that it is threatening the mp3-player. We humans see our mobile phone as a music player, and we are now ready to see it as a music creator. We humans can join forces and create something unique, something new. We can collaborate, through our mobile phones.

- User communication

  The users involved in the creation process need to communicate. Through the MiniBands application the only way will be asynchronously, with text. The created sound will of course be transferred to the next user as a sound. But, the usage of text to transfer short messages seem to be a good fit.

- GUI

  It has been found that the GUI of the mobile application should use an informal GUI without the use of musical notation. This means that the application is made available for both users without any musical training as well as advanced users. Since the average mobile phone user of today uses a mobile phone without touchscreen and various external sensors the interaction with the application will be done through the keypad. The interaction can be extended on devices with higher abilities, for instance through the use of a touch screen or sensors such as an accelerometer.

- Creation process

  The mobile application will support asynchronous participation. One user do not need to collaborate directly, but can wait until it suits that particular individual. This means suitable both in time and space. As with regular music creation, it should support creation in solitude.

- Affordances and Constraints

  The application should minimize dependencies, such as a required bit rate, that can hinder spontaneity. If users are to be able to share their music, something they have spent time with, the system has to prove to be trustworthy. Along with this trust also comes a trust to keep the privacy of the users. The system should be designed so that errors and mistakes are hard to make but easy to recover from.

- Determine application type

  It is important to determine what kind of creative music application that will be created. And, where the application will run. This can narrow things down and make important decisions easier such as what interaction technique to use.

– Give the user incentive

  Some users have to be given an incentive to collaborate. In the MiniBands concept users have to collaborate in order to reach a final piece of music. Another incentives could be to earn money by creating tunes that other people might find interesting to buy, which will create revenue for the creators creating parts of tunes, e.g a market for beats.

2.5 Web applications for mobiles

The following chapter aims to explain technologies that can take the MiniBands concept to the next stage, the mobile stage. The web-bases portal was developed in Action Script 3, a language that mobiles today cannot compile or render. The technologies presented in this chapter will explain how a different version of the application, one written in a mobile friendly language, can be sent to those devices who identify themselves as mobile phones.

2.5.1 Mobile Problems

Today mobile phones are not only limited by small screens, having poor processing power and low bandwidth. They also have to deal with the fact that web content today is generated by a broad spectrum of different programming languages and media formats. Newer phones are of course more adapted to this. They can for example handle applications written in Flash lite (a smaller version of flash running action script 2) in addition to Java (which most phones developed since around 2004 can execute). Older phones with web capabilities only process HTML and nothing else.
Limited screen size is one of the biggest problems with web mobility. If a regular cellular phone had the possibility to compile and run every programming language, the same as a laptop has, viewing the same site on such a limited display would involve a lot of zooming, panning or scrolling to make any sense of the material presented. Therefore data (or an application) that is created mainly to be viewed from a device with unlimited power and screen size (laptop or desktop) must be specially adapted (or split up) to fit a smaller screen.
In this thesis the application that we have created is mainly targeted at material accessed over the web by other devices than phones. We used the Flex framework to develop our content in Action Script 3, a language that phones have yet to figure out how to run. Minibands is however still on the prototype level and in the future we imagine that we implement the idea using Java (or action script 2 for flash lite) to better fit different mobile phones. The processing powers on handheld devices are also limited, and transcoding algorithms for content adaptation often need powerful processors to quickly (and comfortably) adapt content to screen size. This is also effected by the lower bandwidth of these devices. Transferring the whole content from a server for adaptation on the device itself can be time consuming if the content is big. It will also demand a lot more of the bandwidth and the processor of the device and today´s mobile phones are limited in just these areas.
Mobile users today have varying hardware and operating systems to the extent that one solution cannot reach every user on the market, applications must therefore be smart enough to adapt to varying hardware and operative systems on different platforms. This problem is generally referred to as heterogeneity in hardware. If this is not possible

then the application could be created in different versions, each supporting a different hardware (or operative system), this however is time consuming and will demand a lot of memory space for all the different versions. With this strategy the developer has to carefully consider what platforms the intended targeted group for the application uses, and exclude some platforms, reducing the market penetration for the application.

The different approaches to modify web content to fit a mobile environment all use some kind of transcoding. By transcoding is meant the direct digital-to-digital conversion of one encoding to another. This is often done to incompatible or obsolete data in order to convert it into a more suitable format. Transcoding in general means first decoding data to an intermediate format in a way that preserves the original content and secondly encoding the intermediate data into a target format. This process often results in loss of some data. The researcher Björn Knutsson says[4]: "Transcoding is lossy: while it preserves essential information, it removes ("distills") inessential or unrenderable information, in order to meet goals such as bandwidth reduction".
Transcoding can also be a power consuming process, depending on the amount of data that needs to be decoded and encoded. Therefore it is not always optimal to do it on a wireless device. The process of content adapting can take place at three different loci: server, client or intermediate (proxy between server and client). This section will discuss the advantages and disadvantages around these three different loci for transcoding.

It will also, to a certain extent, present some of what standardizing organs and big corporations' solution for this problem is today and what they have planned for the future.

2.5.2 Developing web applications for mobile devices

The term mobile devices include a lot more than just mobile phones. Products like notebooks, sub-notebooks, tablet PCs, PDAs, handhelds and smartphones all fall under the same category. This study will focus mostly on handheld devices with limited display size and resolution since other devices will have displays that are able to access and properly display the main application situated on the web without adaptation.
We are assuming the devices in focus have either GSM data transfer protocols (GPRS) capability or can access wireless data networks. As previously mentioned there are two "main" approaches to developing a web application. First, the version specific approach **(approach 1)** with a modified version of the application for each platform. This approach requires an identification of the platform in order to determine correct application. Second there is the adapting transcoding **(approach 2)** ("write once, run everywhere" paradigm) that can adapt content to any platform with only one version of the application. Implementing different applications for every platform involves extensive reprogramming and therefore might not be realistic to implement given the desired differentiated market penetration. An application that adapts to all different platforms seems like an impossible approach considering the range of different hardware and operating systems on mobile devices. [18]

The web-based portal we developed during this thesis is written in Flex, Adobe's development environment that uses Action Script 3. Mobile phones today are not compatible with this language and this presents a problem since the mobile interface is one of the main selling points of the concept. Flash is currently implemented on some mobiles, under the name Flash Lite 3.0 and using ActionScript 2.0, and it is mainly intended to deliver media such as video and animated wallpapers and small applications. Is a full

flash application implemented in Flash Lite the answers? It would have been possible but since Flash Lite doesn't support mp3 manipulation at run-time it is not considered a suitable alternative for the MiniBands concept.

We did however develop the application with the mobile aspect in mind. Since the application is meant to be run both on computers and mobiles (the whole music creation process is designed so that the interaction fits on a mobile phone). Furthermore, our application is developed using different components. Each component handles a specific task, one for example handles logging in and another playing music. In a computer environment we have a couple of these components on screen at the same time. But, on a mobile we imagined that only one of these components would be displayed at a time. The graphics is also very basic and animations are kept to a minimum, all this in order for the application to work as good as possible on a mobile platform.

Jochen Müller [16] however suggests a combination of approach 1 and approach 2. The idea is to fill in the gaps left from the adaptive approach (approach 2) with the version specific paradigm (approach 1). We choose this approach but when realizing it we focus mainly on the adaptive application approach. The adaptation can be done on the server side, in a proxy or on the actual phone device. Considering the limited memory, computing power and bandwidth that comes with a mobile device it is apparent that it should currently not be done on mobile phone. The adaptation process for the server can be split in two tasks: identification, and customization. The identification task identifies the mobile device and its characteristics such as processor performance and display size. When the server is aware of the capabilities of the device, it can determine whether the device needs a modified version of the application or not. If the device does need a modified version of the application the server adapts the code to fit the performance of the device. This is the customization task.

Identification

The adapting process starts when a phone tries to connect to a web content situated on some server. It does this by sending a HTTP request to that specific server. The first step in adaptation is identification. This process is done where the adaptation process is located (server or proxy). Identification of the device can be done using any of three different methods [16], each with their own success rate. If one fails then the next one is initiated. However more steps result in a less detailed information about the client.
The first method uses a W3C standard for expressing capabilities and preferences of clients and users called CC/PP (Composite Capability / Preference Profile), this will be explained in the next section. An implementation of this step is the UAprof, described in section 2.1.2. If this fails the second method kicks in, this method uses the HTTP Request made by the client. The HTTP Request contains partly optional fields (like accept-, accept-language- or user agent (UA string)- string) which are device specific in some cases. The strings in these fields are compared to a database to determine which device the request came from. Example of matches in the database:

Should also this approach fail a third method is initialized, this step also uses the fields from the HTTP Request. It tries to isolate certain keywords in these fields to get the main characteristics of the mobile device, like display size, processor performance etc. The user agent string in the HTTP request in particular contains these keywords.

type	UA String
Siemens S55	RequirelessWeb/3.2 S55
Pocket PC	Mozilla/4.0 (compatible;MSIE 4.01;Windows CE;PPC 240x320)
Nokia7250	Nokia7250/1.0 (3.14) Profile/MIDP-1.0 Configuration/CLDC-1.0

Figure 2.2: Examples on User Agent strings [16]

Example: take the UA string from the Pocket PC from the example above: "Mozilla/4.0 (compatible;MSIE 4.01;Windows CE;PPC 240x320)". From this string 240x320 can be isolated and used to describe the screen size of the device. [16]

the world wide web consortium and CC/PP

The World Wide Web Consortium (W3C), a massive internet organ developed to standardize interoperable technologies (specifications, guidelines, software, and tools) to lead the Web to its full potential, has developed CC/PP (Composite Capabilities/Preference Profile). CC/PP is a technology that attempts to standardize the process of identifying devices in web requests. The CC/PP is meant to be implemented on devices, so that they can identify their capabilities and their user's preference properly when requesting content from a server. The CC/PP profile will be sent with the HTTP request to the server. [6] The CC/PP (as well as the UAProf) is based on Resource Description Framework or RDFs (a meta-data object created by W3C), RDF provides interoperability between applications that exchange machine-understandable information on the Web i.e. supplying computer readable semantics to regular data. (RDF represents metadata as models that consist of a collection of statements about resources. Statements consist of a subject, a predicate, and an object, where subjects are always resources and objects are resources or literals.)

In an interview on CC/PP by WaSP (Web Standards Project) the W3C replied "With CC/PP, a user with a specific preference, or disability-related need can clarify that even though their browser handles millions of colors, they personally can only distinguish certain colors. Or, perhaps the user navigates exclusively with a keyboard or stylus". We can also imagine that adaptation process will cost money for the user or for the network provider. In this case the network provider (or the user depending on who pays for the adaptation) could lock the personal preference to certain values, so that it will not cost so much. CC/PP is meant to do the following [36] [13] [6]:

- Server-driven content negotiation, this is a version of server-side content adaptation, which will be discussed later
- On-the-fly content selection and presentation based on user agent detection and scripting languages. This results in a smart kind of transcoding based on the user's personal preferences.

Services using the CC/PP can already do the following [36] [13] [6]:

- HTML object and link elements have mechanisms defining alternate behaviors that can be used differently on different devices identified with CC/PP

- SMIL (pronounced "smile"), the multimedia language for audio/visual content, has a switch element defining alternate elements to chose from, and can be used, for example, to choose some content based on available bandwidth that the CC/PP profile will declare to the server

- CSS also has such a mechanism called Media Queries for selecting appropriate style sheets to fit the devices identified with CC/PP

CC/PP profiles will trigger content adaptation, allowing web content to be accessed through a broad range of devices. The transformation mechanism is not yet fully realized by the W3C, but there are many visions and some partly implemented strategies of how to use the CC/PP for content adaptation. Some are already realized by others, like UAProf launched by OMA (more on this in section 2.1.2). [18]

OMA and UAProf

In 2006 OMA (Open Mobile Alliance) released their standard device recognition. It is called UAProf (User Agent Profile) and is a widely accepted standard for describing device performance and user preferences. UAProf seeks to interoperate with the emerging standard CC/PP (from the W3C) distribution over the internet. Since the UAProf is based on RDFs (metadata tags developed by W3C), its development is closely tied to the development of RDFs. In short, this means that the UAProf should adapt to any changes that the W3C does to their RDF framework.

So what actually happens is that the client device sends a request containing the clients' capability and preference information (CPI) through a network towards the origin server. Proxies and routers along the network can add information to the CPI. This is especially interesting if that specific element (proxy or router) can transform the content to a more suitable format. If the origin server or some element along the way that has been made aware of the UAProf capability responds with a specific value of a header this means "OK". This signals to the client that the content will be adapted to its specific CPI. If the client does not receive this "OK", it automatically assumes that the gateway does not support the client's CPI.

The CPI contains (but is not limited to) [3]:

- Hardware characteristics (screen size, color capabilities, image capabilities, manufacturer, etc.)

- Software characteristics (operating system vendor and version, support for MexE (mobile execution environment), list of video and audio encoders, etc.)

- Application/user preferences (browser manufacturer and version, markup languages and versions supported, scripting languages supported, etc.)

- WAP characteristics (WML script libraries, WAP version, WML deck size, etc.)

- Network characteristics (bearer characteristics such as latency and reliability, etc.)

2.5.3 Customization - adapting web content to mobile user agents at different loci

After the identification is completed the customization phase will start. Depending on the computing power of the device different versions of applications are generated. If the device is a PDA or a handheld it has more processing power and better connection possibilities than that of a mobile phone or a smartphone, but still it has a limited display. This yields two versions of the application (in addition to the standard application for desktops and laptops): one **medium** version for devices with more computing power and bandwidth and a **mini** version for mobile phones and smartphone. [18] After determining the general capability of the device and which of these versions to use, the adaption process starts. Depending on whether there is metadata in the source versions (that can guide the content adaption) or not and where the content adaption is located (different loci - further described in following section), this process can take on a number of shapes.
If there is metadata in the source then the content adaption will follow this metadata to adapt the content like a CSS but for different characteristics. Adapting web content that does not have any metadata to fit a mobile device is highly interesting and can be done at different locations (loci). This section of the thesis aims to describe some attempts to do just this on the different loci that are used today, namely on the client, on the server or on a proxy between the client and the server.

Client-side content adaptation

The process of content adaptation can be done on the client side. Client-side adaptation software usually has access to the capabilities of the device. Some approaches to client-side content adaptation are independent of content directives (tags/metadata in the content explaining how the content should be adapted). This can have consequences on the layout intended by the author. Other approaches use directives in the content to make the adaptation. One example is using CSS to style a page. Having two separate files, one for styling and one for content, has its advantages. It lets authors (or users) design different styles to suit different devices. This area is not very researched, since it is currently considered a bad idea to use precious battery and computing power on the mobile device to make the adaptation.
The researcher Suhit Gupta [29] uses a DOM-tree (Document Object Model) to represent the HTML structure of a page. From this tree different filtering techniques are applied to remove and adapt specific nodes and leave only interesting content. These filters are turned on and off by the user in order to customize the content to the users platform and its performance. This technique is often used in "addbuster" software to remove advertisement from regular web sites. [29]
Researcher Oskari Koskimies [17] at Nokia Research Center believes that client-side content adaptation will solve problems such as:

1. No stress is put on the server: no need to encumber already busy servers with the task of content adaptation.

2. Less data in the request: since the device is doing the transcoding, minimal amounts of data about the device's characteristics will be sent in the request.

3. Self aware device: the device knows its own properties and characteristics. It can therefore perform a more precise adaptation than a server that has to rely on

standardized profiles that might be out of date.

4. Real-time customization: the device has a direct link to the user and can ask him/her questions in unclear cases, if the user wants a shorter, high usability or a fuller, low usability version of the content.

[17] [26]

In a not so distant future we can imagine that handheld devices have more processing power, memory, higher screen resolution and longer battery life. This means that mobiles could do exactly what desktops and laptops do today. The need to fit content to smaller screen sizes (not resolution) will however still exist since it will be impractical to carry around big screens wherever you go. When making data portable we can picture the following scenario: Let's say I am working on a presentation on my laptop that I will present later at work. Before heading off to work I synchronize my laptop with my cellular phone i.e. copying the presentation onto my handheld device. In the subway on my way to work I remember some changes I forgot to make on the presentation. How will I access this big graphical presentation on my small handheld screen to make these changes?
Well here some sort of content adaptation could be made in the handheld device to enable it to present and manipulate the data on my limited screen. Since the future will hold cheaper (in battery cost i.e. electricity) algorithms for content adaptation and longer battery life for devices. We could also consider the use of a foldable OLED screen to present the data that is located on the handheld device.

Example on client-side content adaptation:

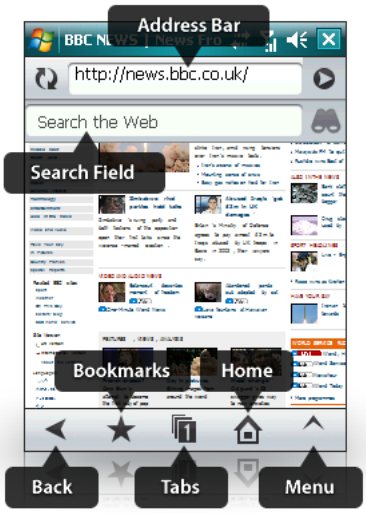

Figure 2.3: opera [27]

"Opera Mobile puts a full web browser on your smartphone. This means that all the rendering is done on the client side so you can access interactive web services including ones using AJAX, and you get added features like tabbed browsing and more!" - www.opera.com

It should however be said that Opera Mobile is made to be used on more powerful mobile units. For regular mobile phones they have another browser called Opera Mini,

that utilizes an intermediate content adaption approach, with a proxy involved (more on this approach in the section on intermediate content adaptation).

Intermediate content adaptation

This type of adaptation is usually applied by proxies and it can be done without any metadata connected to content from the server. However, it is better if the author provides metadata or hints how the adaptation process should be made (if metadata or hints existed, the application would fit in section 3.1 developing web apps for mobile). The process without hints is typically based on transcoding heuristics; the proxy follows certain rules in the adaptation process. These rules consider the structure of the webpage and importance of different elements in the page. Some heuristics base their rules on common designs characteristics of websites. Where links are usually located on a page or where on the page the most important information is usually presented for example. To some extent the tags in the HTTP code can be utilized to find certain useful information, like the header of a page or what text is linked and so on. [18] With

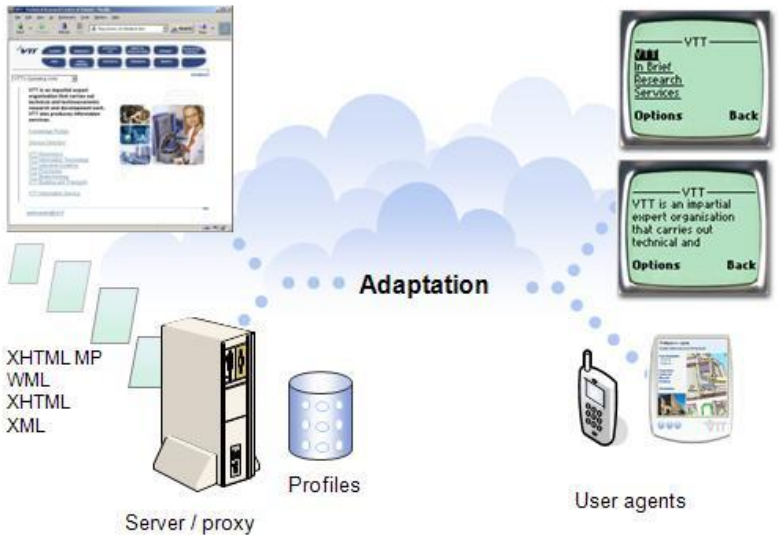

Figure 2.4: Adaptation [18]

more complex websites the task is exponentially more difficult. Wai Yip Lum and Francis C.M. Lau from University of Hong Kong approach this problem by implementing a decision engine. The engine knows the users preferences and the devices capabilities (including bandwidth), with this information it can make up for the lossyness of traditional heuristics since it knows what parts of the content is important to the user. The decision engine is a complex machine that negotiates content in two steps, preprocessing and real-time processing.
Preprocessing includes: data-type and modality analyses. These analyses results in a score that is represented of a specific node in tree of possible renderings of the whole

page. The real-time processing starts from that node and goes to the closest one that the device can handle. It also chooses from four different negotiation algorithms when adapting web content for mobile devices: scored link list (SLL), ordered-relation score tree (ORST), nonordered-relation score tree (NORST) and the score linked list-nonordered relation score tree (SLL-NORST). **SLL**, **ORST**, **NORST** or **SLL-NORST** are all different approaches to finding the best matched node in the tree of possible renderings of the site. [25]

(A context-aware decision engine for content adaptation) Another solution is implemented by Bill N. Schilit in the mLink [5] middleware proxy system. It works with links, websites viewed in a mobile interface using mLink will be shown as a collection of links from that page. The user can traverse this collection until something useful (or not) is found, the interaction with the link takes on different forms depending on what it links to. Links that the client cannot handle (like pdf's or mp3) can instead be sent to the users (home-) computer as emails. (Web Interaction Using Very Small Internet Devices)

Example on intermediate content adaptation: A project called **SMIL** is an intermediate content adaptation that uses a proxy to convert multimedia content to fit smaller devices . SMIL is being implemented on handheld and mobile devices and has also spawned the subset known as Multimedia Messaging Service (MMS) which is a video and picture equivalent of SMS. It defines markup for timing, layout, animations, visual transitions, and media embedding, among other things. SMIL allows the presentation of media items such as text, images, video, and audio, as well as links to other SMIL presentations, and files from multiple web servers. SMIL markup is written in XML, and has similarities to HTML. (Wikipedia accessed 2009-jan-07) What SMIL can do with movie documents for mobiles can be seen on the figure below. [19]

Figure 2.5: SMIL [19]

Server-side content adaptation

Server-side content adaptation is where the server itself does the actual transcoding of the content and then routs it back to the device. End-to-end semantics are well kept in this approach since the server intimately knows the content and can therefore make good (correct) judgments on which parts are more important and which are not. Just as the intermediate approach the "heavy lifting" is removed from the inferior capabilities of the device (client). Unlike the intermediate approach, the servers who do the

transcoding themselves are more prone to overload and delays when they are busy.

Researchers Björn Knutsson and Honghui Lu[4] proposes that server directed transcoding (SDT) be integrated into the HTTP protocol with a compatible and simple extension. Their approach is not a refined server-side content adaptation, it uses directives from the server to perform the transcoding on a proxy. However the transcoding process runs as part of the origin server application, and executes remotely from the origin server solely as a performance optimization, not for any functional reason (the transcoding could just as easy be executed on the server or on the client but that would put too much strain on the server or the client and make the transcoding process slow). [4]

Example on server-side content adaptation: Openwave has a packet of different widgets that together form an end-to-end strategy. The packet includes content delivery technology, MediaCast, and its Adaptive Mobility personalization suite of products. The Nokia S60 smartphone platform supports widgets and web developers can use this platform to create widgets for other Symbian phones using the S60 operating system. Another part of Openwave is OpenWeb and secure content management. Product strategist Ed Moore says it has the capability to deliver full web pages to any data-enabled device. OpenWeb uses a server element for content adaptation, and has the capability to adapt standard websites to fit mobile environment. OpenWeb was licensed to Vodafone in Spain in 2007.
According to Moore the Openwave research shows that usability is not necessary the most important element for mobile web access. Instead he claims that users want to access same sites they do on their PCs on their phone, and once users have this access, mobile-to-web access will increase with up to 25 %. [31]

Figure 2.6: Openwave, as pictured in Wireless Week Jan. 08. [31]

2.5.4 Analysis

Limits to screen size, battery life, processing power and bandwidth, how can we approach these problems that are tied to today's devices? This section will propose one possible

solution based on the technology available today and discuss how it would access most web content using different techniques and standards presented in this chapter. What will the future look like? What technologies are being developed for tomorrow? This discussion will also attempt to give some insight to how these problems will be bridged in the future.
In this chapter/thesis a few different techniques for content adaptation are suggested, and with additional metadata a lot of the already existing web content could be accessed in a comfortable way from mobile devices. However the adding of metadata to suit all sites is not at all something that is practical. What could be done is that whenever a request is sent to page that is without metadata (or other mobile device adaptation friendly approach) from a mobile device, a proxy (like those discussed in chapter 3) initiates a translation or adaptation of the page to fit the device. The adaptation (or rather the metadata created) is then saved as an addition to the page. So that the next time someone tries to access the content from a mobile device it can use the previously created adaptation. If the site is created with mobile compatibility this will be no problem. But where can this additional data be stored? A proxy will not have authority to manipulate sites all over the world (how would that look?). I suggest that this could be solved by a simple add-on in the browser connected to a proxy (that creates the actual data) and a database that links the added data to the specific site. This might however consume a lot of space for one single database as the project runs on and more sites are connected to the "system" (browser add-on + proxy + database = system). One solution to the space problem would be some kind automated author contact process. After metadata has been added for a site, a request to update the site could be sent to the author of the page. After an update has been done, the metadata would no longer exist in the database. Instead the adaptation would be done directly from the website.
Even with 3G network bandwidth on mobile devices some media formats for movies (and high quality sound) still takes a lot of time to download and is heavy for mobile processors to process.
In the future, when 4G wireless networks and OLED foldable displays are realized, bandwidth and screen size will not be a problem. In that future scenario only the processing power, battery life and graphical hardware will set the limit to the content that can be displayed on a mobile device. Maybe adaptation will be completely unnecessary if your mobile device and your computer are equally powerful.
Jinlin Chen at Microsoft research [15] in china claims that his system, Adaptive Web Content Delivery (AWCD), is more robust and extensible than other systems for content adaptation. AWCD however does not approach mobile phones instead client devices for AWCD include Windows CE based Palm-size PC, Hand-held PC, Laptop, Desktop. And in a future where a mobile is equally powerful to these devices it can be the way to go.

2.5.5 Guidelines

The best approach today for making web content available for mobile devices is of course to program the applications as smart as possible to make them device independent. However there is already millions of websites that are not created with this in mind, therefore additional techniques should be applied to handle these cases. The technologies discussed in chapter 3 are all different approaches to this. While mLinks (Schilit) is adapted to older gsm-phones, Wai Yip Lums approach is better at adapting web content for more modern devices.

What about devices that can't be identified? These devices are considered too old to be expected to have a functioning internet connection. The automated response when a phone cannot be identified could be to send the most primitive version using only HTML to give the device a chance to read the content. If you want better adapted web content, you better get a new and better phone.

A combination of smart programming and a few different techniques (Schilit and Lum combined for example) should yield an almost platform independent web access. If these techniques were to be collected under one proxy and added to the system suggested in the discussion above, it would be a powerful tool for mobile access of web content. In the future of our application we imagine that our mobile interface will utilize the different components that we compose our interface in a scaled version to fit mobile screens.

The mobile web of tomorrow however will probably be doing transcoding (scaling) in the phones themselves. We can already see this emerge as we watch Iphone´s Safari browser or other smartphones (or PDAs) using Opera Mobile browser. These browsers are made for mobile units with more processing power than that of a regular mobile phone. And in time average Joe will carry a phone with enough battery life, processing power and bandwidth to comfortably access the web in any form, shape or size from his pocket phone. However interaction methods and screen size will still be issues that will need dealing with. Maybe with ocular implanted screens, foldable OLEDs, brain to computer interaction or clever voice interaction will solve these issues.

Chapter 3

Design process

This Chapter describes the development process from brainstorm to prototype.

3.1 Method

We were assigned the task of creating a prototype for an application for music collaboration, which in the next stage would become mobile. An early request was that both interfaces should support the same interactions and that the interaction of the music creation process was to be easy enough for beginners yet efficient for experts. An initial meeting with a mobile software developer resulted in a few simple scetches of how information and content would be distributed to and from desktops and mobiles, see figure 3.1 - 3.3. The technical specifications of the concept that were to be created was at however still quite thin. To structure what was supposed to be created and perhaps find new solutions, a couple of brainstorm sessions with participants from both the developer team and the owners of the project were performed. These sessions gave a clearer view of what was expected and some deeper technical specifications.
Even though the mobile interface was supposed to be a later step in the development process we had to find out the limitations of this not yet existing mobile application in order to create a web interface with the same capabilities. In order to complete this task we had to choose a platform for the mobile application, and to find the one best suited we had to look at the mobile phone market of today and guess what they would look like in upcoming years. Should this application be created for mobile phones or Smartphone's? What interaction methods had to be available on the running device, what capabilities had to exist and what limitations could be critical. User studies and a lucid scenario were created to provide a look and feel of the intended mobile application. The scenario was also used to find out how the parts of the system would communicate. Besides the mobile application other important agents such as server, portal and database were incorporated into the scenario. This scenario was basically a way to understand the task of each agent and to get a visual on the flow of information. When user studies, scenarios and technical specifications were completed the blue prints of the system architecture could finally be created. It includes what platform to use, what classes, methods, database tables and connections that has to be implemented and a sequence diagram that shows how information is exchanged between the layers of the application. With this information at hand the implementation process could begin.

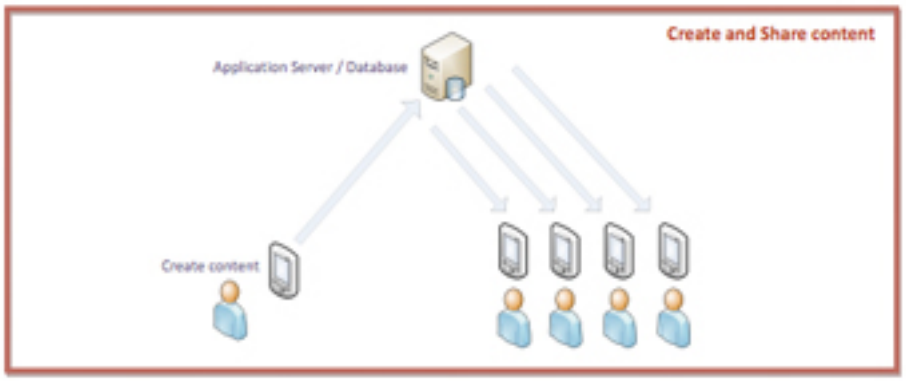

Figure 3.1: Create and share content

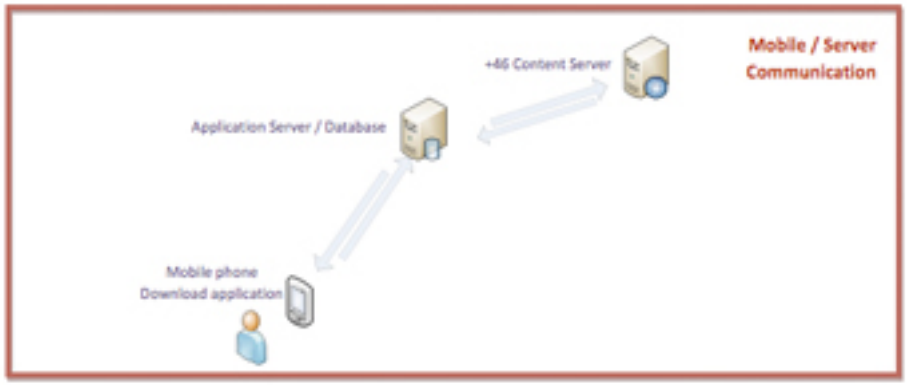

Figure 3.2: Mobile/server communication

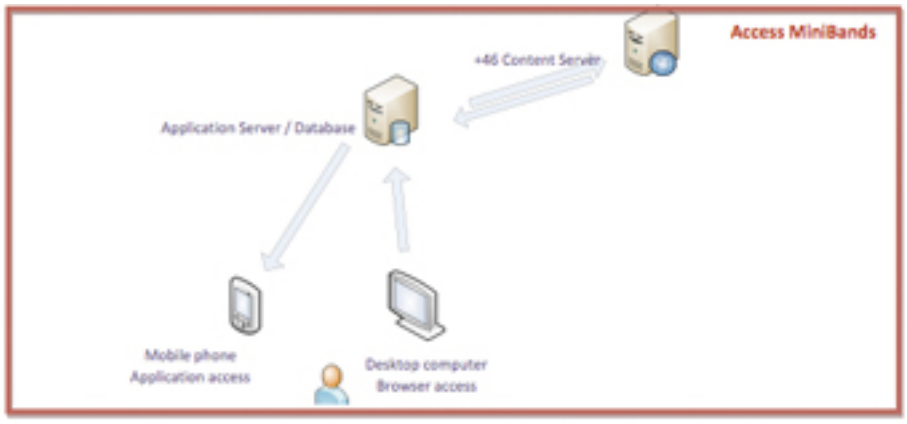

Figure 3.3: Access the Minibands portal, both mobile and stationary

3.2 User studies and technical specifications

When we started the project the idea was pretty rough and not very detailed. Brainstorm sessions was therefore performed together with representatives from the company that

initially were involved with creating the mobile interface of the system. The brainstorm was conducted with one secretary that wrote down the ideas which emerged during the session. After the sessions were finished the findings were discussed that lead to a more thorough specification of the possibilities and limitations of the intended system from a mobile point of view. Still, specific scenarios were lacking. To specify this user studies were performed. These user studies, or user oriented scenarios, were loosely restricted. Each participant started a scenario and then sent it on to the next person to try out what would happen in the music creation process when users made different choices along the way. The ongoing activity was documented and information gave a pretty detailed vision of the system and what technology that should be used.
The user studies resulted in a detailed scenario, see appendix A figure A.1 - A.3.

The graphical interface design was initially very much up to us as developers to create. We asked ourselves as users and developers to find out what features we wanted to implement on the page. A lot of the functionality was taken from standard practices from other web communities and portals (like Facebook or Myspace) one example was making the Minibands logo in the header component a link back to the index page. We wanted users to feel at home directly and never feel stuck in the navigation. Incorporating deep-linking with swf-address in our application made it easy to navigate and always allowed the user to change the state of the application wherever he/she was. Deep-linking is a solution to solve the lack of state-specific http addresses in flash. and swf-address is a technique that allows deep-linking. We also came up with the idea of making the design in components to make the interface easily scalable for different devices. A while into the project a designer helped out with the graphical design and feel of the interface. Given a few colors, graphics and fonts we had a lot to go on to get the application consistent with his visions.

3.3 System architecture

When we first reached To implement the requested features a structural process began. Different diagrams and basic structural schemata were designed and modified to fit the process at hand. After the retrieval of important user information by examining user scenarios, personas and human-computer-interaction guidelines a cross check between the findings and the technical specifications of the project took place, see Appendix B for these findings.

The findings were important when it came to deciding on what technical solution to choose. The main technical request that almost by itself was the factor of deciding determined which web platform to use was the fact that the portal had to be able to support real time music creation. Among the web platform techniques available the only one that could support this and still support other requirements such as integration with the browser, ability to display animated and interactive objects was Flash. Since the real time audio manipulation library only was available for Action Script 3, the targeted Flash Player was set to 9, the lowest player version supporting this new language.
As mentioned above the server had to be able to communicate both with the mobile client and the web portal. When the decision was made to use Flash as the web platform, the server was chosen as an AMFPHP server that works well with Action Script, but also share communicative abilities with the java implementations of the Action Message Format (AMF).

3.4 Implementation

Since the system created would have to work with two interfaces, or rather two applications, a three-tier architecture was implemented where the second layer, the data communication layer was located on a web server. The two applications could therefore use the same data communication layer, but specific data handling were implemented into each application.

3.4.1 First layer

During the first stages of developing the portal the built-in state feature in Flex was used. This feature is used to control the state of the application. After a short time of development it was found that this route was impractical since the system was not able to access variables from previous states and therefore lost track of important information. The state approach was replaced by a component-based design pattern where every component has its own connection to the middle layer (Server) and can manage its own states and variables. Another feature in using components is that when the application is adapted to mobile interfaces the components do not need to be redesigned to fit, instead a well selected, limited amount of components can be viewed with a limited screen size, compared to many components at all times from a computer interface. If another component is needed the user will simply swich states to view that component.

Components can communicate between each other when they need information or variables, for example the user profile component communicates with the login component to learn what user is logged in at that moment. Also, these components were easily created in the Flex development environment and from the Flash IDE (Integrated development environment). Even thought the use of components created a more complex system it made the system easy to maintain and allowed components to be moved, updated or even removed without causing the system to break.

3.4.2 Second layer

In the second layer the data communication was structured and developed to fit the intended actions of the first layer. The communication between the first layer and the second layer was created using a PHP implementation of the Action Message Format (AMF) known as AMFPHP. AMF allows for binary serialization of Action Script native types and objects that are to be sent to server side services. Since our application runs on Action Script this was a natural choice. The portal was as mentioned above created in Flex, but the mobile application platform was yet to be determined. The choice of using the AMF was however considered as a good one since also java applications can make use of it through a Java AMF Client.
All communication with the database was done in PHP, which is an interpreted scripting language that allows a web server to create dynamic web pages, or in this case web services. The PHP server scripts talk directly to the database server that in this case was a MySQL server. The PHP web service solution was used since Flex does not communicate with the database directly; however, it can consume data from a web service and display it.

3.4.3 Third layer

The database was one of the first things that got implemented in the process of building the system. During the phase when the system architecture was to be determined a number of database structure charts were created. This was done in order to see how information was to be stored and how everything should be connected and structured to work optimal. In the database implementation no stored procedures or automated relational connections was implemented.

Chapter 4

Results

Minibands.com is the prototype portal of the MiniBands system that the project ended up with when the deadline for the thesis was reached. It works as a normal community/portal where every user has his/her personal page. Every song created belongs to a band and its members. The portal is, to some extent, open to visitors other than users, which is an attempt to open up for spreading the created music. This means that a non-user can listen to songs created by users. All listeners might not be interested in participating in the creation process. This yields that the portal has two different interfaces (on certain components) depending on whether you are a visitor or a user (logged in or not).

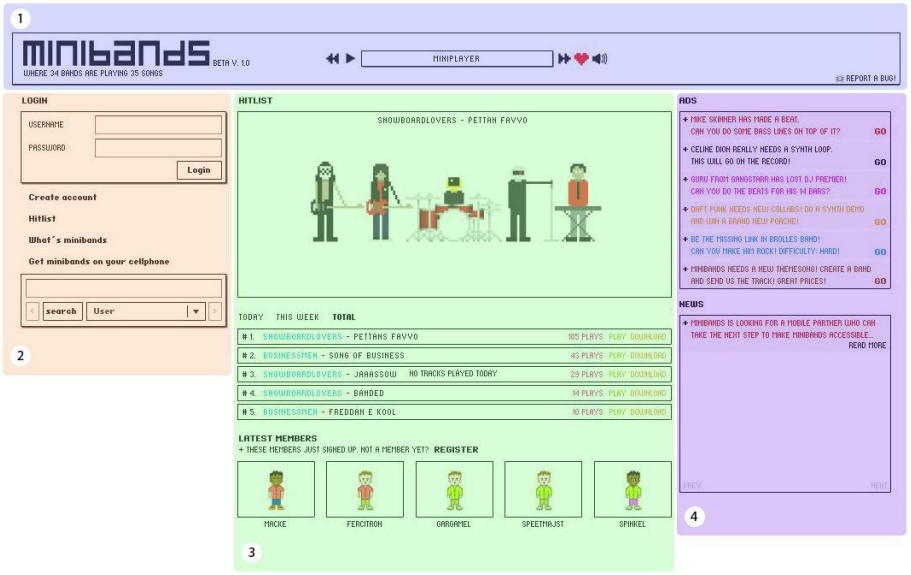

Figure 4.1: Overview of the Minibands web-portal

The first screen on the application contains a **Header** (1 in figure 4.1) that handles the music playing (and acts as a link back to the main page), it also contains the Minibands slogan, and there is also a button where users can report bugs in the Minibands application to Minibands administrators. This component is static and stays with the

application regardless what state the user is in.

Other than the header, the system is divided into three sections We will refer to them as **Left section** (2 in figure 4.1), **Middle section** (3 in figure 4.1) and **Right section** (4 in figure 4.1). In terms of usability this is a common way to implement a portal. By doing this we get a natural focus and layout of the application that is intuitive and builds on user knowledge about (other) portals. The components are placed under these sections and ordered according to their importance. This design contributes to our usability goal (stated in the problem specification section in Chapter 2).

The **Left section** (Figure C.1 and C.2 in Appendix C) is somewhat static, it will change only a few things depending on the state of the application. It contains a **login component** that handles user login and logout and keeps track of what user is logged in. Furthermore the left section contains a **link component** which has clickable links that differ depending on if a user is logged in or not. In the logged out state, the link component contains four links:

- Create account - to access the full potential of Minibands, users need to be logged in to their account. To get an account they click this link and follow the steps of the account creation process (more on this on the middle section under create account component). This link is not available when users are logged in to the system (no need to create an account when you already have one).
- Hitlist - This is a link back to the mainpage, so that the user never gets stuck in the navigation on the application.
- What's Minibands - This links to a page that describes Minibands.
- Get Minibands on your cellphone - A link for users or visitors who wants to know more about Minibands mobile application.

When a user logs in to the application more links pop up in the component(and the "create account" link disappears).

- Me - a link back to the users profile page (navigational purposes)
- Inbox - links to the users messagebox, where the user can read and send messages
- Invites - expands the invites on the user profile component in the middle section
- My bands - expands the bands on the user profile component in the middle section
- My songs - expands the songs on the user profile component in the middle section
- Make song - links to the music creation interface

At the bottom of the left section there is a **search component**. The search component allows users and visitors to search Minibands for songs, users or groups. The result of a search is presented under the search fields in the search component. The results are clickable, and transports the user/visitor to the results component. This is a user profile component if the search is of users, or band profile component if the search is of bands. Clicking a song, will however just play that song in the header. More on these components is found in the middle section below. The contents of the left section are

motivated from studying other portals. Having login and navigational links placed here throughout every state of the application is easy for users to understand and use.

The **Right section** (Figure C.3 in Appendix C) is static to users. It contains two components the ads component and the news component. The **ads component** is where Minibands will present competitions, prizes and winners.
Below the ads component, the **news component** is found, where news on the Minibands application development is presented. Both these components are updated by administrators of Minibands. These components are seldom relevant to users and by placing them to the right they are not in the users focus, unless he/she carefully traverses the page. They do not change whatever state the application might be in and does not draw much of the users attention.

The **Middle section** is the section that contains the most important information. It will display a different component for every different state of the application. In fact the middle section is where specific information, function and content of each state is presented. Between the middle section and the header is a text describing what state the application is in , i.e. what component is showing in the middle section, to let the user know where he/she is.
First out is the **mainpage component** (Figure C.4 in Appendix C), which is the first thing visitors/users will see when they access the Minibands page. It presents the most popular songs (sorted by day, week and total), most recently registered members and also acts as a stage where the band plays when music is played on Minibands by the header component.
When a user logs in he/she is directly transported to his/her **user profile component** (Figure C.5 in Appendix C). Here the user can see his/her avatar, pending song invitations (expandable), bands (expandable) and songs (expandable).
Clicking on one of the bands will transport the user to the **band profile component** (Figure C.6 in Appendix C), where details about the specific band are presented. A collection of the avatars of the users that makes up the band, published songs, a button to start a new song with the same band constellation, unpublished songs, group content like pictures, a button to upload more photos/pictures and a wall where users and visitors can write opinions and messages concerning the band.
When a user wants to create a song he/she can either click "Make Song" in the link component or the "create new song" in the band profile component. Doing this will view the **create song component** (Figure C.7 - C.10 in Appendix C). If accessed from the Link component, the user can create a whole new band, by choosing a new bandname. If this component is accessed from the band profile component, the user can only create a song with the same band members as in the original band. When a song name (and band name) has been chosen the "start creating" button will be clickable. Clicking that button will change the state of the component and allow the user to choose an instrument. When this is done the music creation process starts. When the user is satisfied with the music track he/she clicks "save and continue". The component changes state again and presents the user with the option to choose the next user (friend, phonebook, email, etc) who should play the next instrument on the track (limited to band members if predefined band was chosen earlier). When this is done the "Send" button becomes clickable. Clicking "Send" will send the user back to his/her profile page.
Clicking the "inbox" link in the link component will show the users **inbox component** (Figure C.11 and C.12 in Appendix C). This component shows all the messages that

the user has received from other users and allows new messages to be created and sent (to members from the users bands). Clicking a message in the user inbow will view the message and present a "reply" button. Clicking "reply" will view a message with a re:"original message subject" as subject and the user from the message as recipient. When in the "send a message" state there is also a "clear" button that resets the message and subject. In the future this button should contain a control popup when this is clicked to make sure the user didn't click by mistake.
The **About component** (Figure C.13 in Appendix C) is accessed from the "What's Minibands" link in the link component. It contains information that explains what Minibands is created for and what its possibilities are etc.
A visitor on Minibands.com (not logged in) will be able to click the "Create account" link in the link component. This opens the **create account component** (Figure C.14 - C.16 in Appendix C), Account creation is done in three steps, three states of the component:

1. Personal information (name, phone number, e-mail address, username and password). When this is correct the "proceed to next step" button will be clickable.

2. Avatar creation. Here the users can customize their own avatar. The avatar will later be shown both on the users profile page and on bands the user is part of. When the user is satisfied he/she clicks "save avatar", this allows the user to proceed to the last step through the "proceed to next step" button.

3. The third and last step of the account creation process is optional. It (will allow) allows the user to get the Minibands application sent to his/her mobile phone. To do this the user is asked about the brand and model of his/her mobile phone, so that the correct version of Minibands mobile can be sent to the phone. The user may also choose to get the mobile application later.

Clicking "Get Minibands on your cellphone" button in the link component will send the user to the **Minibands mobile component** (Figure C.17 in Appendix C). This component explains the advantages of having the mobile version of Minibands, and will in the future allow users who don't have the mobile application to get it. The middle section of the Minibands application is where the state dependent data is shown. The users focus when changing states will be directed here and this is where the most important information is focused.

Chapter 5

Summary / Reflections

5.1 Project work

Since we have been situated in Umeå and the company that we work with are officed in Stockholm the workflow during the thesis has been very irregular. When we started the project we did not have many directives to follow, and we started developing to get more ideas. Along the way problems popped up and we dealt with them over phone, skype or email. This could sometimes halt the development process since we might not get a straight answer right away. Problems in the head office in Stockholm also lead to a lot of delays on their side and since we had a time limit set to only one semester the application suffered. In the end we had to set a deadline that we couldn't work past, and at that point in time we would call the prototype done. Working in a project of a small size, the flow is very dependent on some people. If one person is sick, the project might be halted for a week. But since the project group was so small we got to test many different roles and were allowed many freedoms when interpreting given directives. A big part of the prototype is a direct result of our own ideas for the application and not the directives from the company.
We took on a couple of different roles while developing the software. On one hand we studied and developed user scenarios that would later help shape the way that users interact with the software. On the other hand we worked as pure software engineers, designing the software according to strict guidelines provided by the design team (where we also worked).
To reach the goals listed we entered a project with peacock advertising, who had an idea about developing a multiuser web- and mobile- software with the purpose of letting users create music together with other users. Our position in the project was first of all to develop a prototype of the web-interface. This prototype was basically meant to be used to demonstrate the system and its purposes when peacock tried to sell the system and presenting the idea to possible development partners. To accomplish the task more background information was needed, mainly in two areas: incentives for users to use mobile-interface and discovering tools to use when developing multi-platform software. The first task implicates a look at mobiles and why users choose them over stationary means of communication. In the multi-platform development tools area Peacock advertisement wanted us to choose the development environment for the prototype. First we had to decide what our target was going to be. The mobile phones used today have a wide range of compatibility depending on brand, model and year.

Should we try to reach all mobiles or just those with certain technological requirements? Since we did not develope the mobile client we never really got around to answering that question. But we have not limited future development to specific phone features, like accelerometers or touchscreens. such features could instead be considered as extentions to the interaction with the mobile application.

5.2 Fail/Success

Whether the project was a success or not is somewhat hard to say. Depending on what goal you set out in the beginning or what role you were given, it could be both. Our goal as students was to get a working prototype for the web portal and write the thesis report. But as engineers the goal would have been more about getting the music creation process to run smoothly and test it with real users to find its usability. As project managers the goal would have been to sell the idea/concept/prototype/application and start making money off it. As we discussed earlier we were given many different roles and the goals for each of those roles was difficult to reach. We have a prototype that can be accessed on the web (student goal), we did however not have time to do any usability testing (engineering goal). And, the concept of Minibands is yet to be sold to an investor for further development. But if we were to prioritize the goals, we think we have managed the most important one. This is after all a course and we are not yet engineers or project managers. It has been valuable to try these different roles and we are satisfied with what we have accomplished given the available recourses.

5.3 Acknowledgements

We wish to thank the people at Peacock Advertising Agency, especially our external supervisor Marcus Holmlund, for giving us the chance to work with this interesting and challenging concept. A big thank you to our internal- and program supervisor Håkan Gulliksson who has contributed a great deal, not only to this master thesis, but also to the great years at the Interaction technology and Design program.

References

[1] Tonium AB. Pacemaker - the pocket-size dj system. http://www.pacemaker.net/, February 2009.

[2] C.-F. Soerensen A.I. Wang and T. Fossum. Mobile peer-to-peer technology used to promote spontaneous collaboration. Technical report, Department of Computer Inormation Science, Norwegian University of Science Technology, Trondheim, 2005.

[3] Open Mobile Alliance. Open mobile alliance template specification for user agent profile. Technical report, 2006.

[4] H. Lu B. Knutsson and Bryan Hopkins. Architecture and performance of server-directed transcoding. Technical report, University of Pennsylvania, 2003.

[5] D.M. Hilbert B.N. Schilit, J. Trevor and T.K. Koh. Web interaction using very small internet devices. Technical report, Intel Research, 2002.

[6] Dr Mark H. Butler. Cc/pp and uaprof: Issues, improvements and future directions. Technical report, Hewlett Packard Laboratories, Bristol UK, 2002.

[7] A. Dekel and G. Dekel. Mogmi: Mobile gesture music instrument. Technical report, The Hebrew University Jerusalem, The Selim and Rachel Benin School of Computer Science and Engineering, 2008.

[8] L. Denise. Collaboration vs. c-three (cooperation, coordination, and communication. Technical report, The Resselaerville Institute, 2007.

[9] A. Elsdon. Mobile music creation using pdas and smartphones. Technical report, 5th International Mobile Music Workshop, University of applied arts, Vienna, 2008.

[10] G. Wang G. Essl and M. Rohs. Developments and challenges turning mobile phones into generic music performance platforms. Technical report, Deutsche Telekom Laboratories and TU Berlin Center for Computer Research in Music and Acoustics, 2008.

[11] L. Gaye and L-E Holmquist. Performing sonic city: Situated creativity in mobile music making. Technical report, Future Applications Lab, Viktoria Institute, 2003.

[12] Gracenote. Gracenote: Mobile musicid. http://www.gracenote.com/business$_solutions/mobileMusic/, Febru$

[13] J. Liu H. Lin, S. Jiang and s. Lin. Introduction to cc/pp technology. Technical report, IBM China Software Development Lab, 2006.

[14] K. Jennings J. Bligh and B. Tangney. Designing interfaces for collaborative music composition. Technical report, Centre for Research in IT Education, Trinity College, Dublin, Ireland, 2005.

[15] Y. Yang J. Chen and H. Zhang. An adaptive web content delivery system. Technical report, Microsoft research, China, 2000.

[16] D. Henrici M. Hillenbrand P. Müller J. Müller, T. Lenhart. Developing web applications for mobile devices. Technical report, Department of Computer Science, University of Kaiserstautern, 2004.

[17] O. Koskimies. Metadata for client-side content adaptation. Technical report, Nokia research center, 2004.

[18] Timo Laakko. Adapting web content to mobile user agents. Technical report, VTT Information Technology, 2005.

[19] T. Lemlouma and N. Layaïda. Smil content adaptation for embedded devices. Technical report, OPERA project, 2003.

[20] M. Lievonen. Where collaboration takes place. Technical report, Royal Holloway University of London, School of Management, Egham, Surrey, 2005.

[21] Albert Lin. Understanding the market for digital music. Technical report, Department of Economics, Stanford University, 2005.

[22] Harmony Line. Harmony line - compose music online software. http://www.hyperscore.com/, February 2009.

[23] Last.fm Ltd. Last.fm - discover new music with free internet radio and the largest music catalogue online. http://www.lastfm.se/, February 2009.

[24] Spotify ltd. Spotify - a world of music. instant, simple and free. http://www.spotify.com/, February 2009.

[25] W.Y. Lum and F.C.M. Lau. A context-aware decision engine for content adaptation. Technical report, University of Hong Kong, 2002.

[26] R. Gimson M. Butler, F. Giannetti and T. Wiley. Device independence and the web. Technical report, HP Labs, Bristol, 2002.

[27] Opera Mobile. www.opera.com. http://www.opera.com, january 2009.

[28] B. Reimer. Why do humans value music? Technical report, Vision 2020. Reston, VA: MENC, 2000.

[29] D. Neistadt S. Gupta, G. Kaiser and P. Grimm. Dom-based content extraction of html documents. Technical report, Columbia University, 2005.

[30] M. Schrage. *Shared Minds*. 1990.

[31] B. Smith. Are browsers dead? - wireless week january 01, 2008. Technical report, Wireless week magazine, 2008. http://www.wirelessweek.com/Article-Handset-Browser-Use.aspx (accessed 2008-nov-08).

[32] A. Tanaka. Mobile music making. Technical report, National University of Singapore, 2004.

[33] Wikipedia the free encyclopedia. Superdistribution. http://en.wikipedia.org/wiki/Superdistribution/, february 2009.

[34] J. Vincent and C. Gilleard. I just can't live without my mobile! Technical report, Receiver Magazine, 2006.

[35] J. B. Walther and K. P. Daddario. The impacts of emoticons on message interpretation in computermediated communication. Technical report, Rensselaer Polytechnic Institute, 2001.

[36] wasp internet magazine. Introduction to composite capabilities / preferences profile (cc/pp). Technical report, The Web Standards Project, 2004.

Appendix A

Mobile scenario

This appendix shows an intended scenario when the system works on mobile phones.

Figure A.1: Step 1-6 on the user scenario

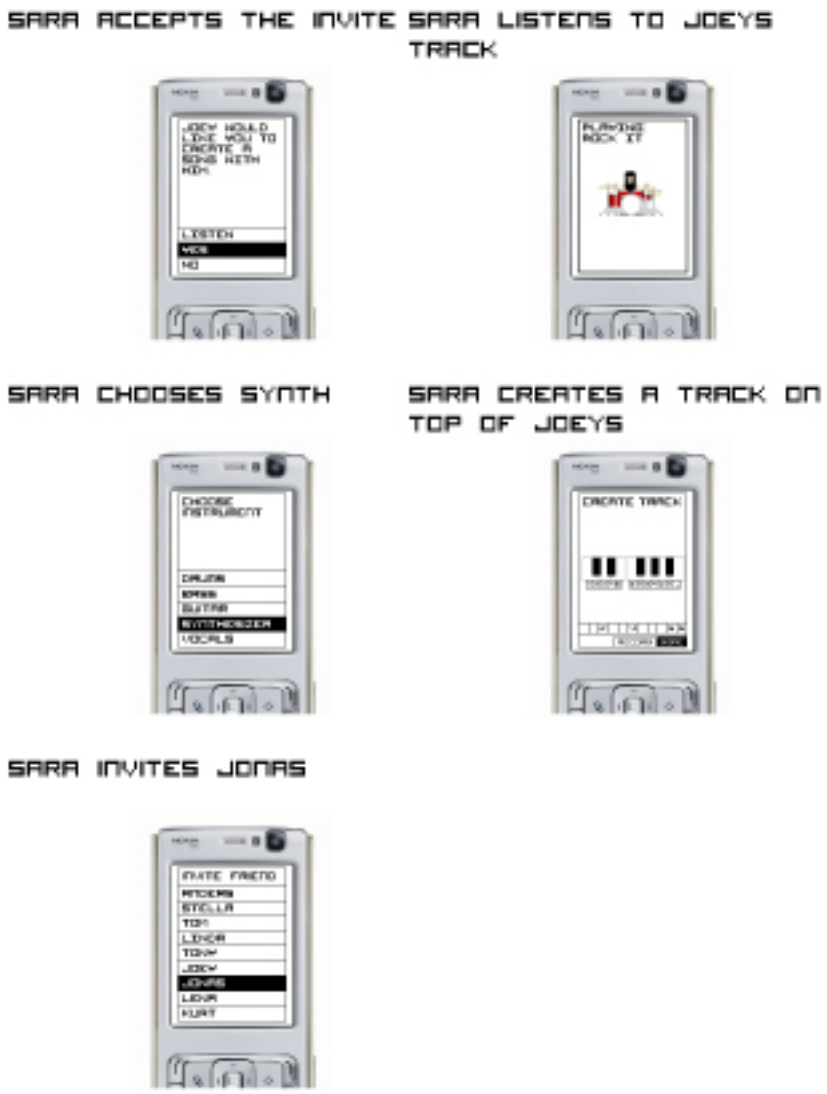

Figure A.2: Step 7-11 on the user scenario

Figure A.3: Step 12-17 on the user scenario

Appendix B

Schemata for the system

This appendix contains flowcharts, classes and database tables that we created to initiate the implementation phase of the project.

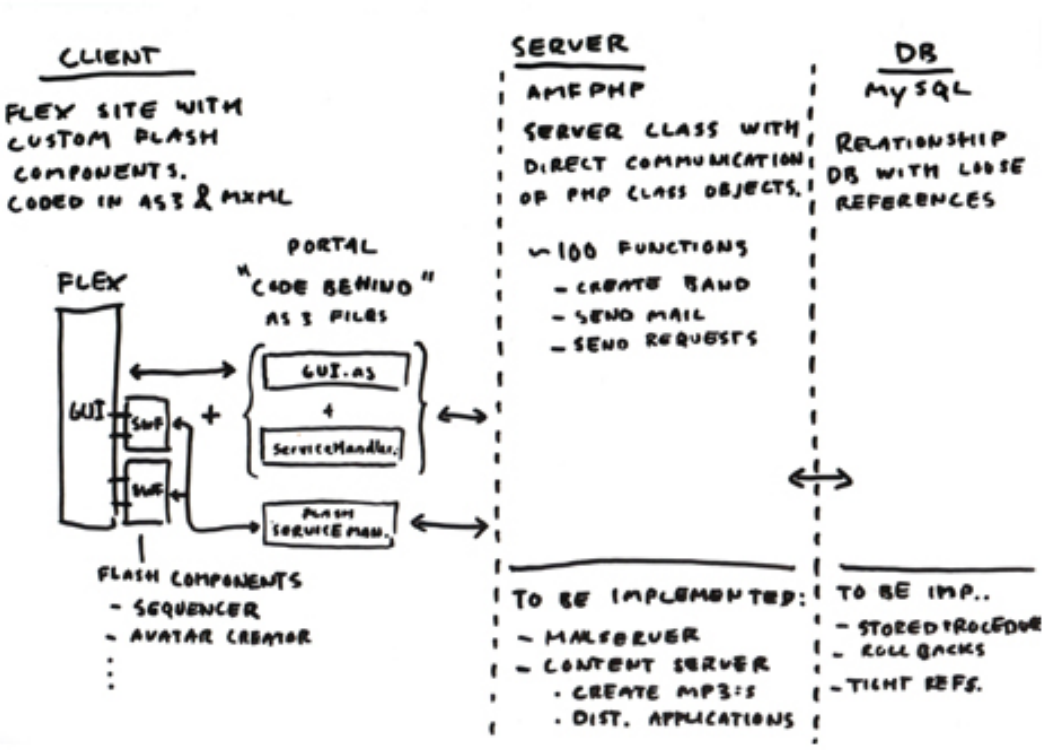

Figure B.1: Early scetch of the system parts and how they communicate

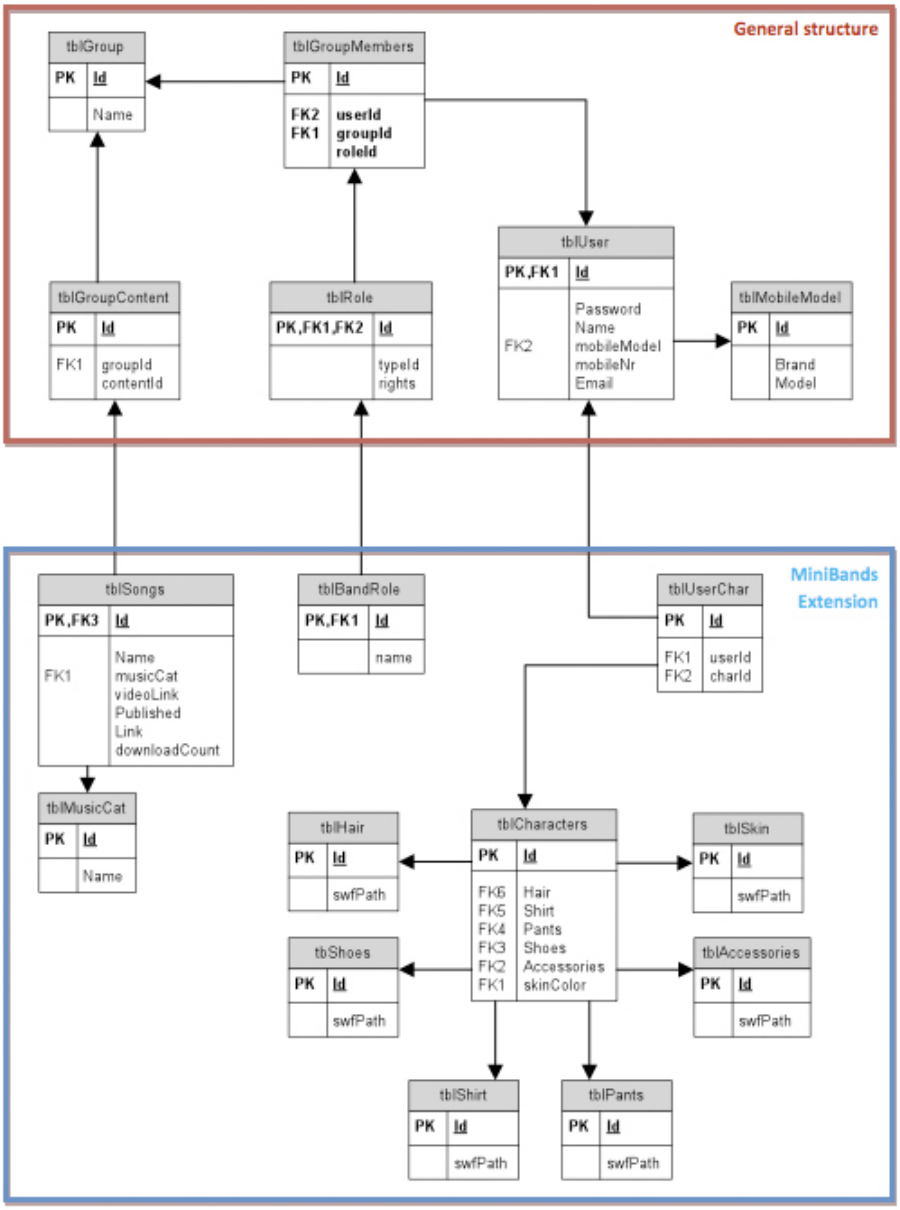

Figure B.2: Data Base tables

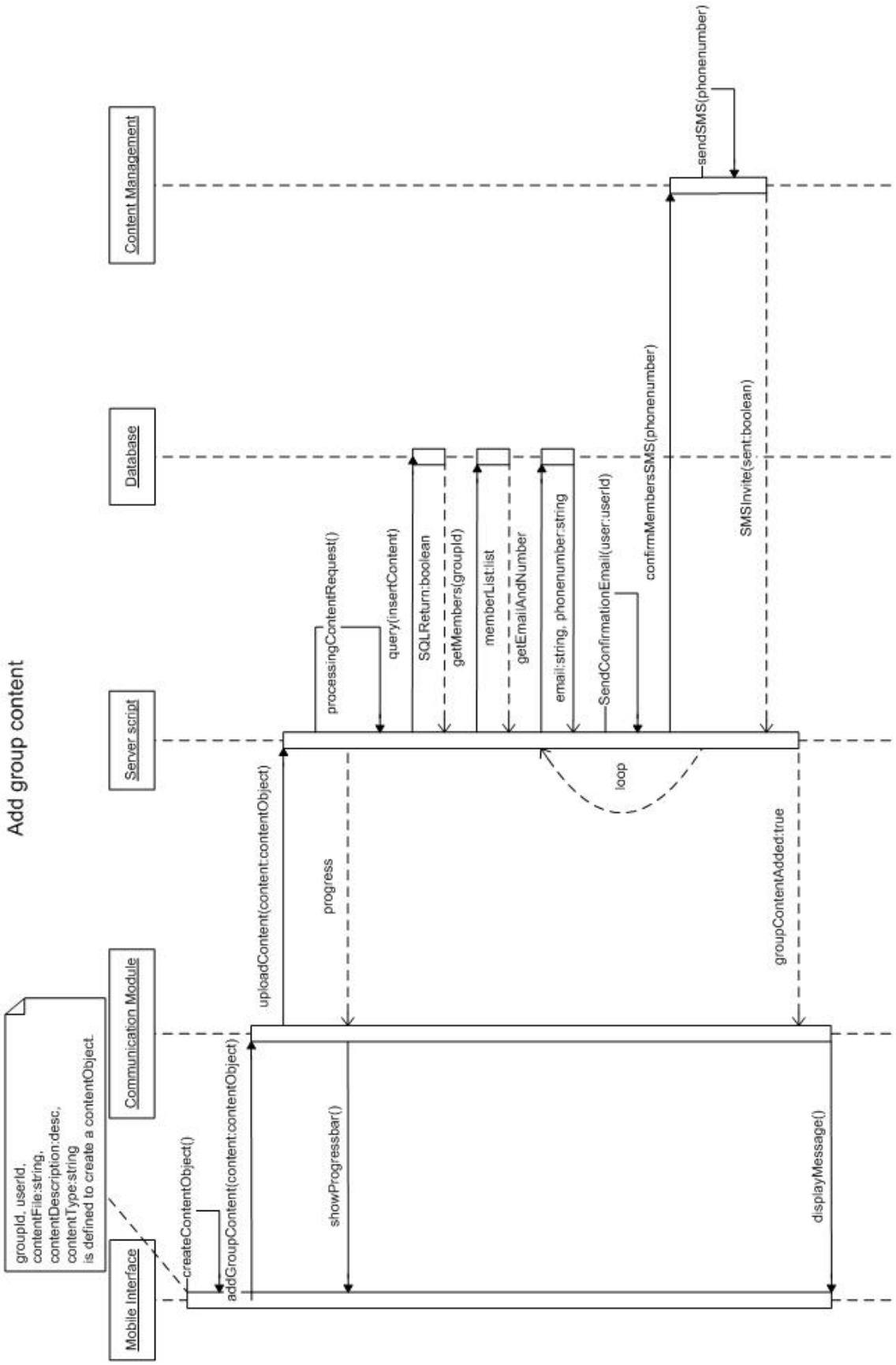

Figure B.3: adding content to a group

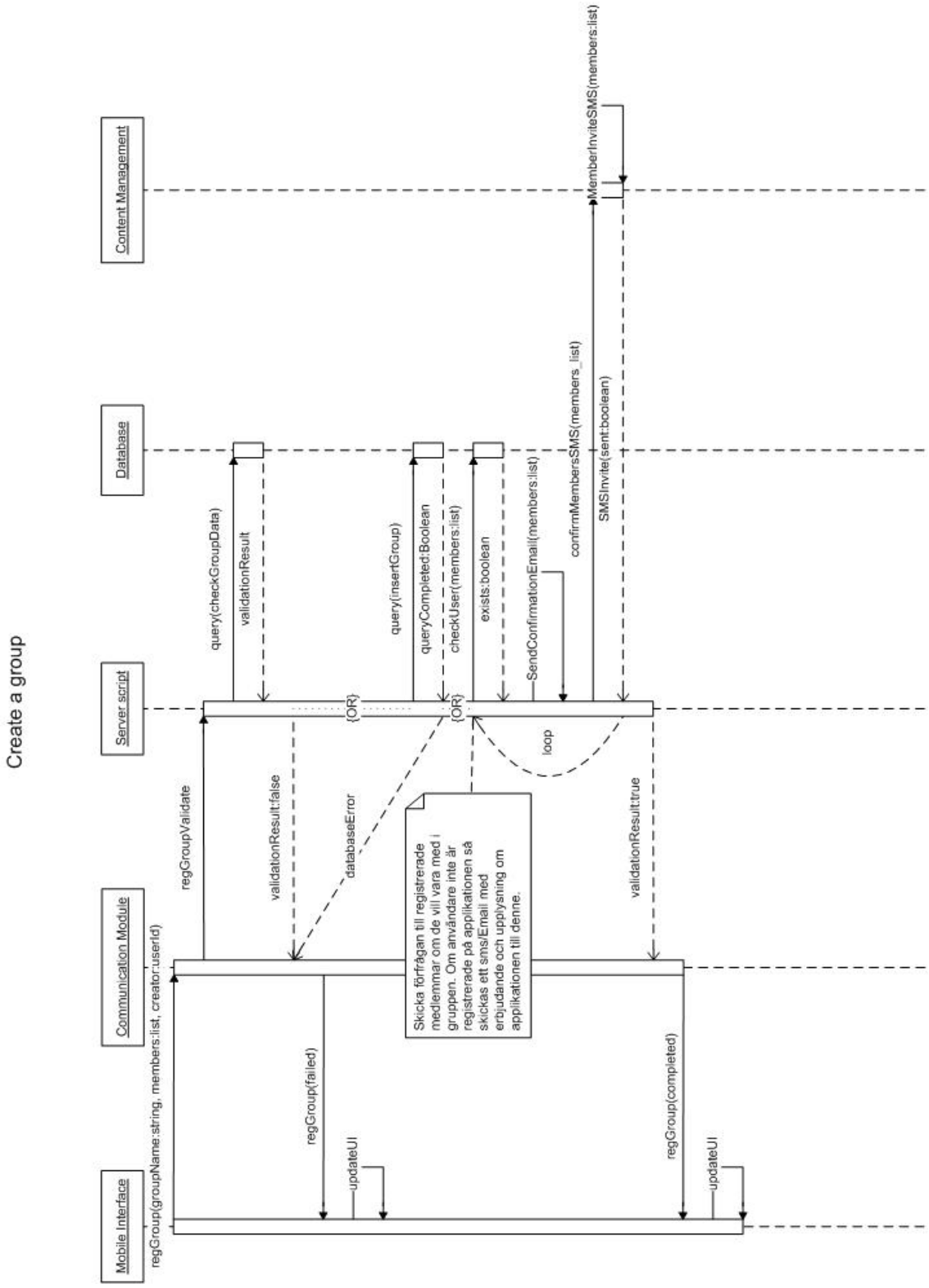

Figure B.4: creating a Group / band

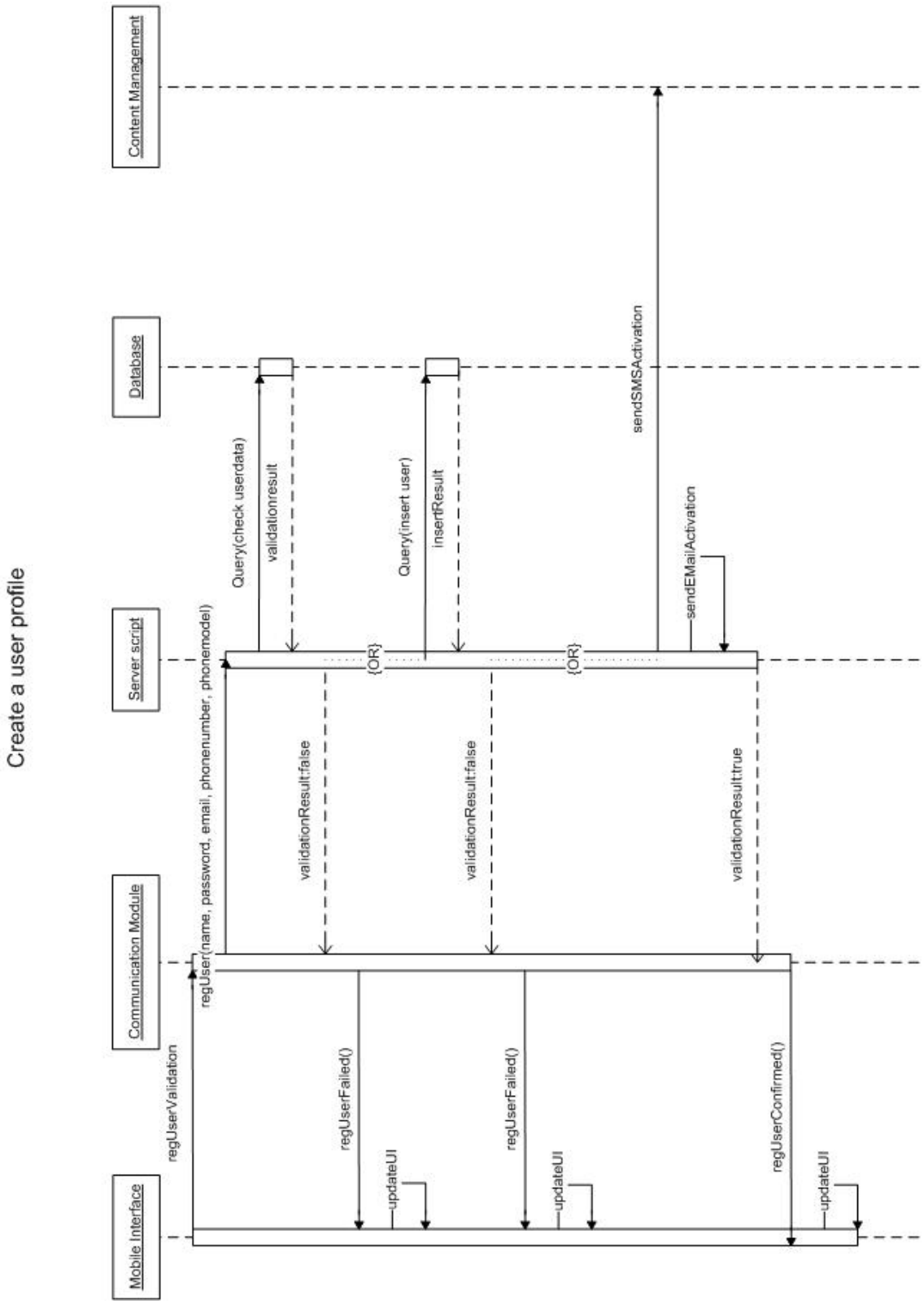

Figure B.5: creating a user profile

Portal Tables

tblBand
- Id (int)
- Bas (userId)
- Synth (userId)
- Drums (userId)
- Guitarr (userId)
- Vocals (userId)
- Name (varChar)

tblUser
- Id (int)
- Password (char)
- Name (char)
- charId (charId)
- mobileModel (mobileId)
- mobileNr (int)
- email (char)

tblSongs
- Id (int)
- BandId (int)
- Name (char)
- musicCat (catId)
- videoLink (path)
- published (bool)
- link (mp3/xml) (path)

tblCharacters
- Id (int)
- Hair (hairId)
- Shirt (shirtId)
- Pants (pantsId)
- Shoes (shoesId)
- Accessories (accessId)
- skinColor (skinId)

tblMobileModel
- Id (int)
- Brand (char)
- Model (char)

tblMusicCat
- Id (int)
- Name (char)

tblHair
- Id
- SwfPath (path)

tblShirts
- Id
- SwfPath (path)

tblPants
- Id
- SwfPath (path)

tblShoes
- Id
- SwfPath (path)

tblAccess
- Id
- SwfPath (path)

tblSkins
- Id
- SwfPath (path)

Sound tables

tblClassic
- guitarPack (path)
- synthPack (path)
- vocalPack (path)
- basePack (path)
- drumsPack (path)
- specialPack (path)

tblRock
- guitarPack (path)
- synthPack (path)
- vocalPack (path)
- basePack (path)
- drumsPack (path)
- specialPack (path)

tblPop
- guitarPack (path)
- synthPack (path)
- vocalPack (path)
- basePack (path)
- drumsPack (path)
- specialPack (path)

Figure B.6: database tables

Appendix C

Screen shots of the actual system

This appendix shows the different sections and components (described in Chapter 4) that make up the prototype that was the result of our thesis.

Figure C.1: left section before login and search

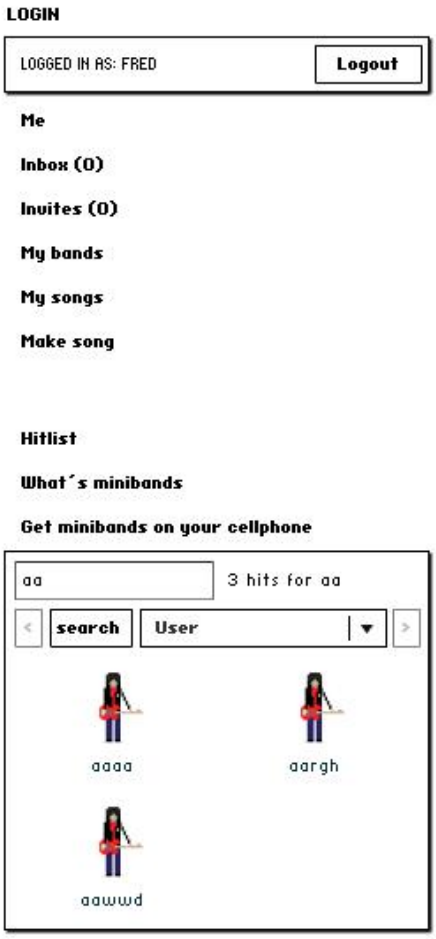

Figure C.2: left section after login and search

Figure C.3: right section

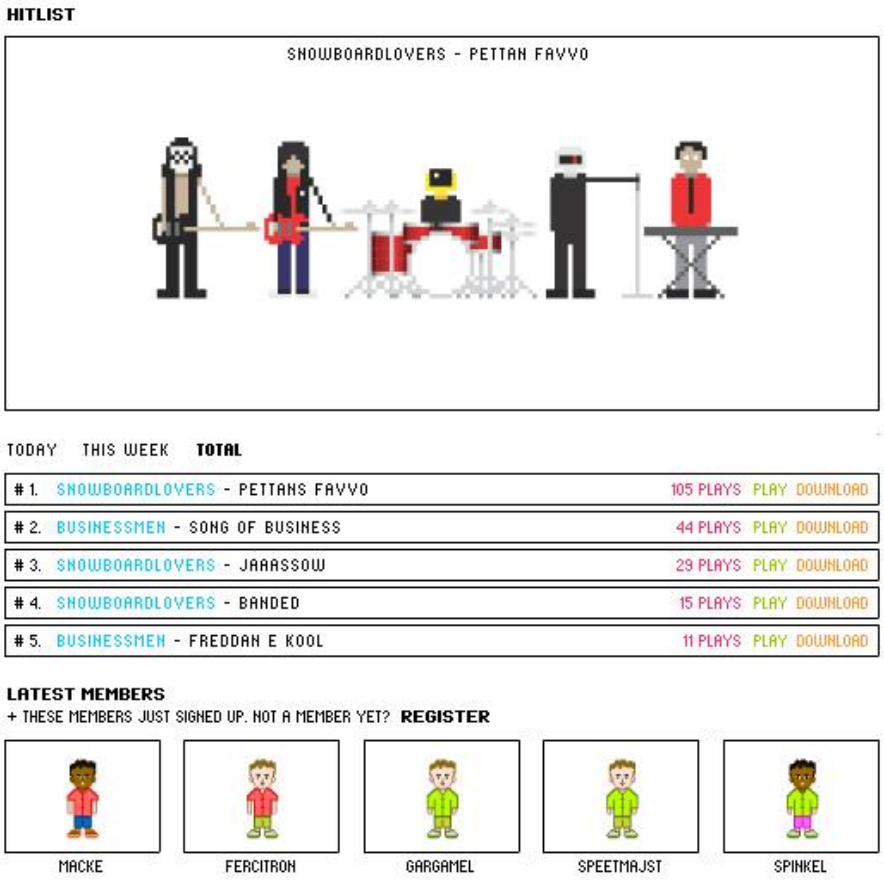

Figure C.4: mainpage component

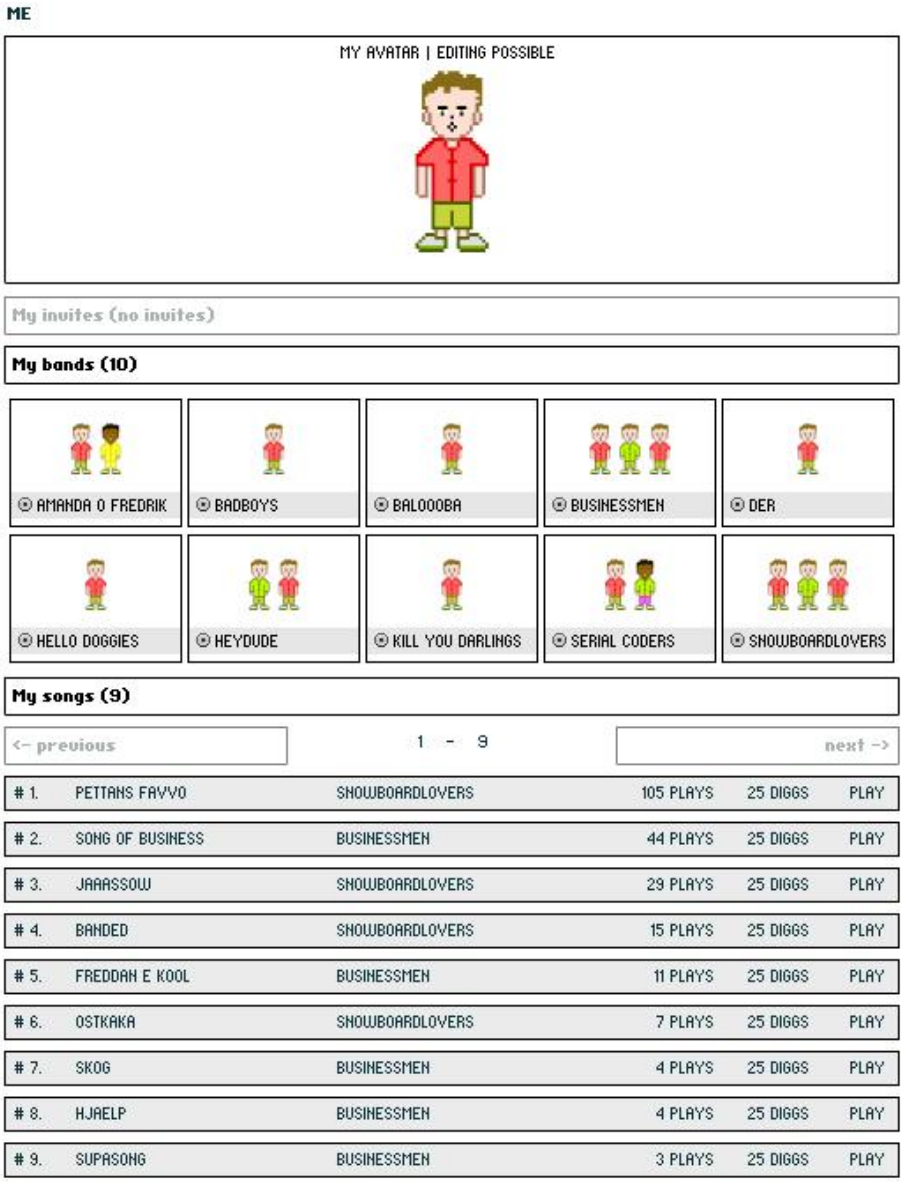

Figure C.5: user profile component

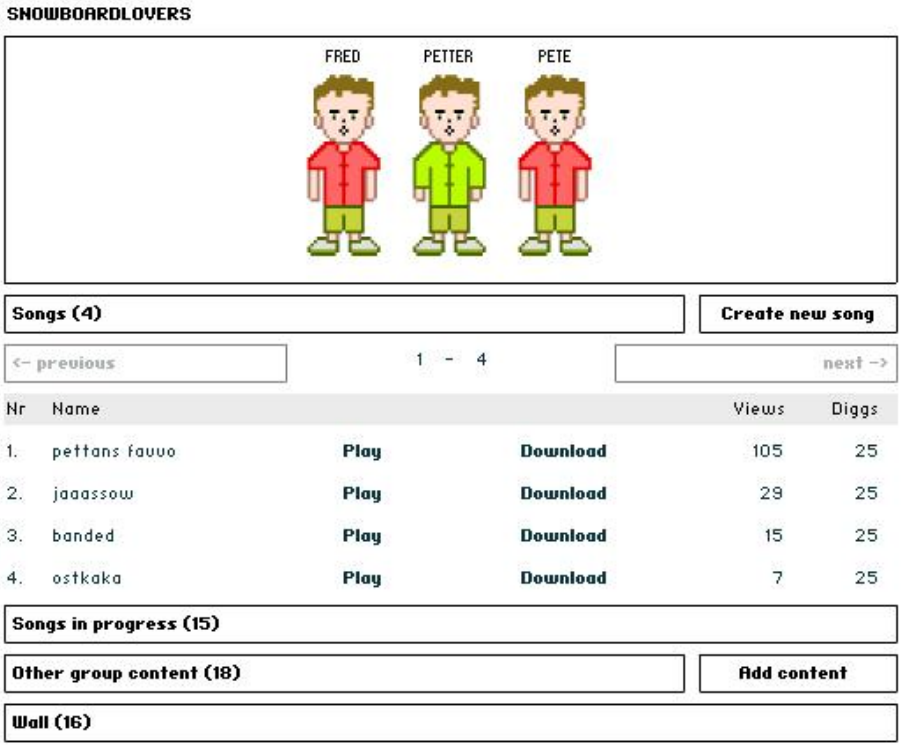

Figure C.6: band profile component

Figure C.7: first step in creating a song

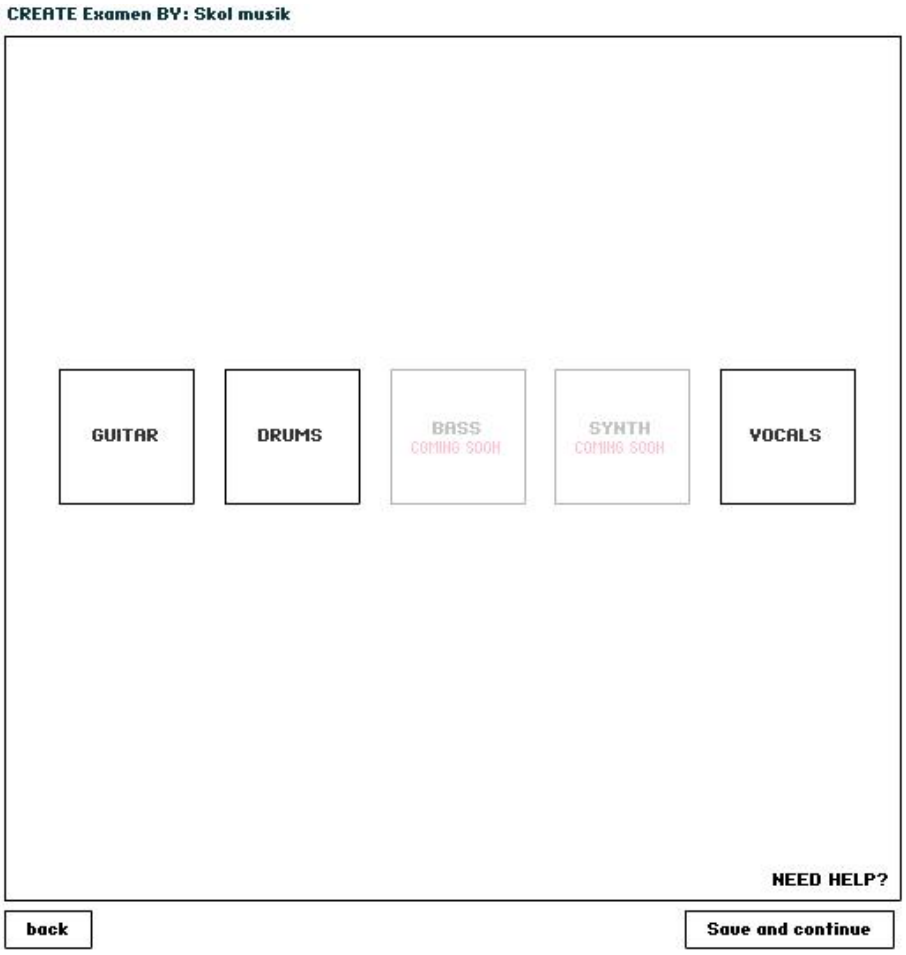

Figure C.8: second step in creating a song

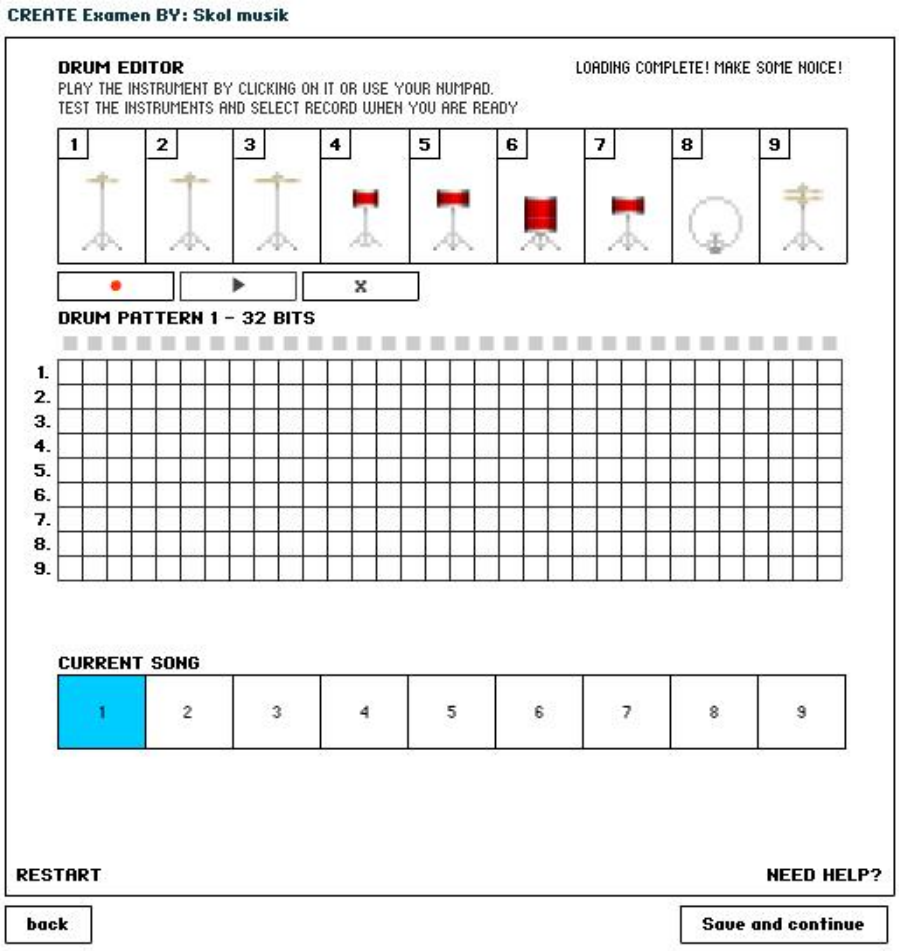

Figure C.9: third step in creating a song

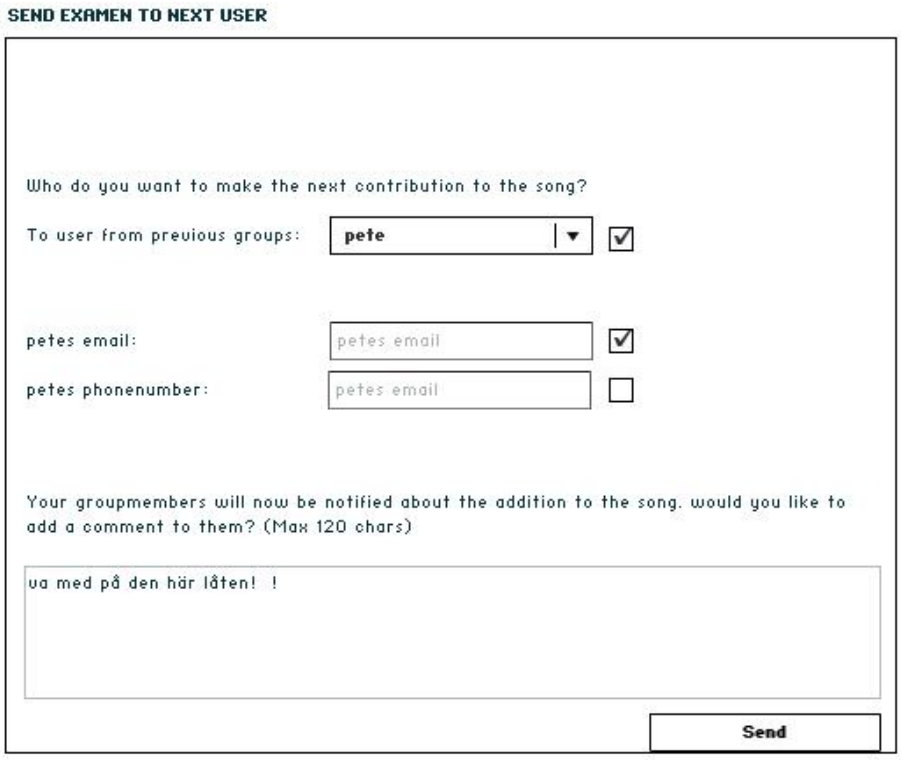

Figure C.10: fourth and last step in creating a song

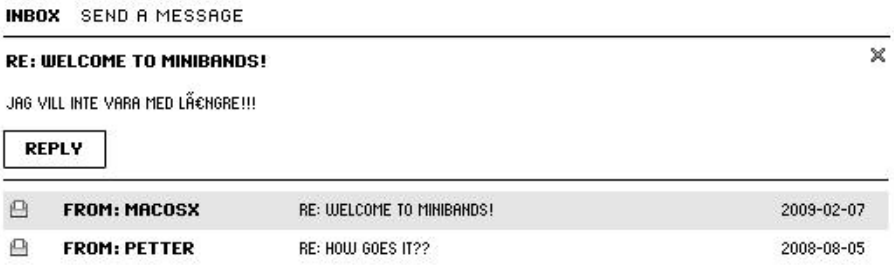

Figure C.11: viewing a recieved message in your inbox component

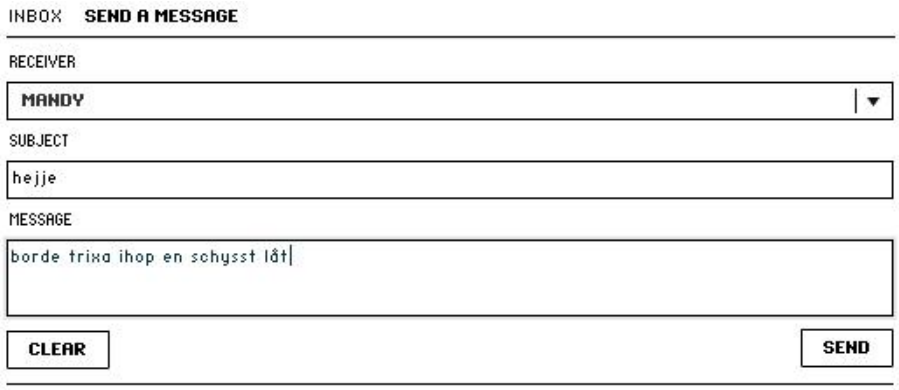

Figure C.12: creating a message with the inbox component

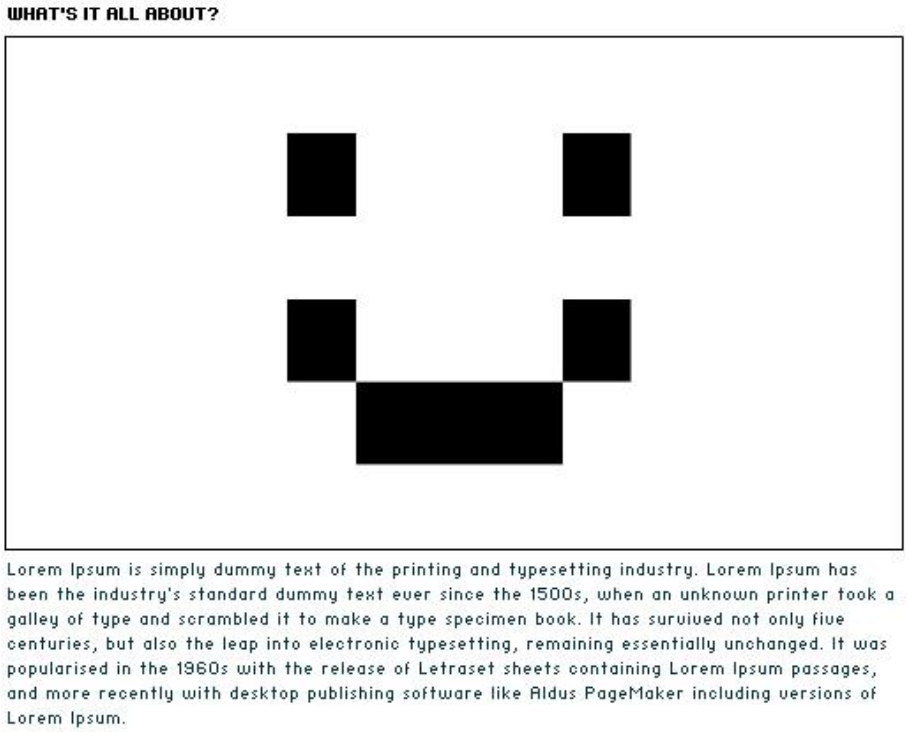

Figure C.13: the about component

Figure C.14: first step in creating an account

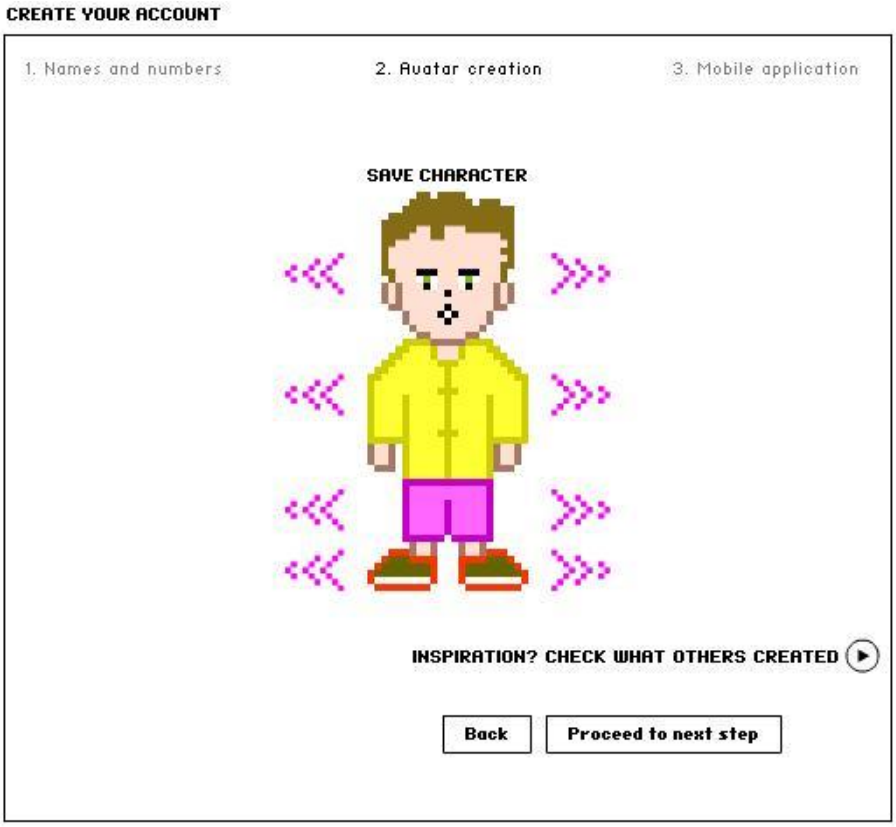

Figure C.15: second step in creating an account

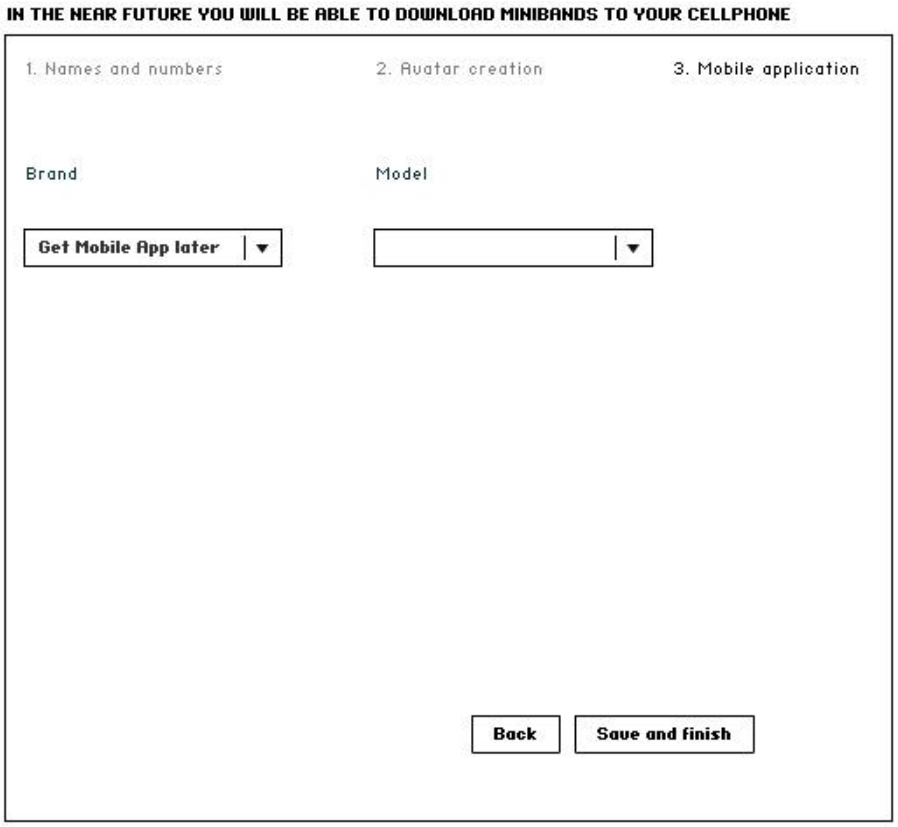

Figure C.16: third and last step in creating an account

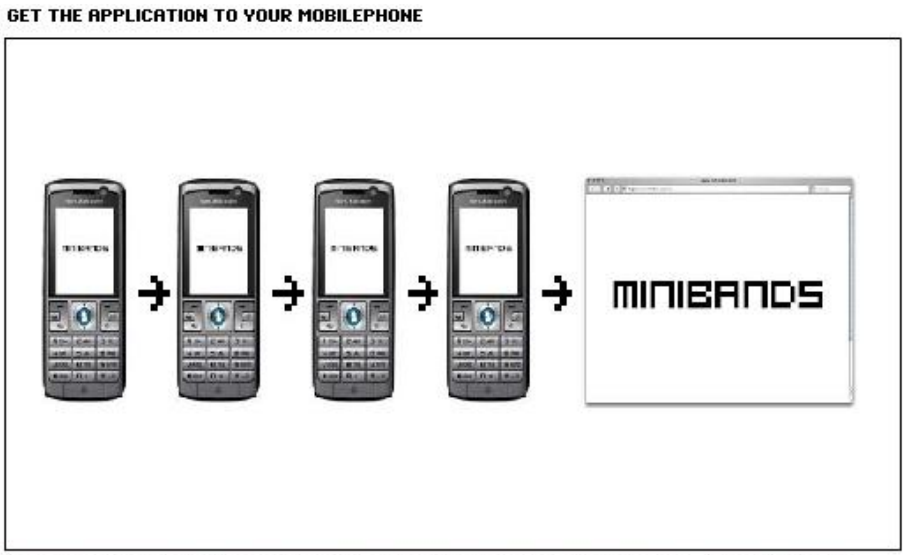

Figure C.17: the mobile component