LiveTV for Mobile Applications
Live TV broadcasting to mobile phones

Version 1.1

Doc. No.:

Revision History

Date	Version	Description	Author
2009-09-28	0.1	Basic information	Darko Ronić
2009-09-29	0.7	Chapters 3-8	Darko Ronić
2009-09-29	0.71	Project leadership defined	Darko Ronić
2009-09-29	0.9	Chapter IX added	Darko Ronić
2009-09-30	0.91	Small additions	Darko Ronić
2009-10-02	1.0	Version freeze	Darko Ronić

Table of Contents

1. Introduction

1.1 Purpose of this document

LiveTV for mobile applications project goal is to provide the technology for live video broadcasting via mobile phones with clients using their mobile phones to view the broadcast. Along with the broadcasting, the system will provide the means to control the broadcast and add various commercials.

This document will describe the project goals, milestones, activities and intended users. It will set a timetable and dates for the team to follow.

1.2 Intended Audience

This project has a wide specter of intended users. Everybody with a mobile phones could and should be able to use this service. It is primary intended to enable broadcasting of events to mobile phones, so everybody who wants to watch the event is our intended user. If we count the people involved in the event recording as users, then we should expand our users to broadcasting company employees.

1.3 Scope

This document is intended for description of project roles, team organization, project risks and milestones. It does not deal with technical or implementation details.

1.4 Definitions and acronyms

1.4.1 Definitions

Keyword	Definitions

1.4.2 Acronyms and abbreviations

Acronym or abbreviation	Definitions
FER	Faculty of electrical engineering and computing, Zagreb
MdH	Mälardalen University, Vasteras, Sweden
SVN	Subversion revision control software
API	Application Programming Interface
S60	Symbian S60 operating system
VLC	VideoLAN server used for streaming

1.5 References

Project Homepage
http://www.fer.hr/rasip/dsd/projects/livetv

Project Google Group
http://groups.google.com/group/livetv_dsd

2. Organization

2.1 Project management

Project will be developed in co-operation between team members from both Croatia and Sweden, with their team leaders and a project manager which will coordinate teams. Project leadership is still discussed because two members are interested in the PL spot.
As of version 0.71, Clément has been chosen as the project leader.

2.2 Project group

Name	Initials	Responsibility (roles)
Clément Fouque	CF	Project leader; Studio application developer (C#); Testing
Amer Tahir	AT	VLC and Studio application developer (C# and C++)
Dalibor Mesarić	DM	Studio application developer (C#); Testing
Nima Moghaddami Khalilzad	NMK	Java player developer (Java)
Darko Ronić	DR	Team leader; Recording application developer (Symbian C++)
Neven Vujasinović	NV	Recording application developer (Symbian C++)
Željko Rumenjak	ŽR	Java player developer; SVN manager

2.3 Steering group

Project implementation is practically opened to interpretation. Customer's opinion is that the team should decide how to implement the system. Since a half-working version of the system already exists, it will be used as a guideline but without any important influence to the project. Regular meetings will be held with project supervisor. Iterations in the development will give us a good feedback to see if our path is correct.

2.4 Customer

Project customers are Damir Isović (vice-dean at MdH) and Klas Eriksson (OneDial AB).

2.5 Others

Project supervisor is Rikard Land from MdH.
Project leadership will be decided via a poll because two team members are interested in project leadership.

3. Assumptions and constraints

3.1 Technological

Project will be based on several technologies, mainly programming languages and existing solutions for different parts of the system. The existing solutions are either open-source or commercial applications provided by hardware manufacturer.

The technologies are:

- Symbian C++ programming language
- C# programming language
- Java ME programming along with MMAPI framework for multimedia
- VideoLAN VLC server for video streaming
- Darwin Streaming Server by Apple for streaming of content to clients
- Symbian S60 operating system with various Nokia smartphones (N95, e66...) and Symbian API
- 3GPP family of codecs (include h.263+ and AMR)

3.2 Environmental

There are four environments that have to be mentioned:

1. Symbian S60 operating system – recorder mobile phones will be Symbian based phones so recording application will work in this environment
2. Linux/Unix operating system – server containing VLC server. May be later discarded in favor of libvlc implementation in C#
3. Windows operating system – Studio application will be developed for Windows
4. Variuos other mobile operating systems that support Java – Java environment for mobile phones. Clien player will be made in Java so it can be run on virtually all mobile phones.

3.3 Interpersonal

Because of geographical distance between two parts of the team, interpersonal communication is limited to online services such as Google groups, Skype, audio and video conferencing. Local parts of team will communicate in person.
Google groups are used as the primary solution for asynchronous communication, while Skype is used for synchronous communication.

3.4 Work distribution

Work will be divided into three parts because there are three parts of the system that have to be developed. Team members are divided so that members of the team are locally together. Since there are three parts, one parts will have geographically divided members.
One part of the team will develop the recording application for Symbian, one will develop the studio application, while the last part will develop the client player application.

3.5 Casual relationships

None.

3.6 Time

The tim constraints on this project are very tight. Not so much about the coding parts but because of various documentations. That's why we will try to follow the deadlines to best of out ablilites.

4. Deliverables

To	Output	Planned week	Promised week	Late +/-	Delivered week
Steering group	Project Description	40	40	0	40
Steering group	Requirement Definition	40			
Steering group	Design Description	41			
Steering group	Revised Design Description (opt.)	43			
Steering group	Acceptance test plan	50			
Steering group	Final Project Report	2			
Steering group	Code and documentation	2			

5. Inputs

From	Required item	Planned week	Promised week	Late +/-	Delivered week
Steering group	Previous project implementation	39	40	1	still not
Steering group	Linux server access	40	40	0	still not

6. Project risks

Possibility	Risk	Preventive action
40%	Deadlines not met	Parallelization of work, and implementation of basic functionality first
10%	Team member drop out	Other members capable (time and knowledge) of taking over additional work
20%	Design oversights	Make sure one person is assigned to some task which requires early involvement in the perceived riskiest parts.
15%	Implementation problems	Redefinition of the project design, different technologies, different approaches

7. Communication

Communication on the project can be divided into two parts. First is the asynchronous communication, the other is synchronous communication.

Primary means of asynchronous communicate shall be a Google group and emails to members. This kind of communication will be used to announce other of some information or achievement.

Synchronous communication will be mostly used on the project. There shall be groups meetings on Skype multi-chat or conference call twice a week (Sundays and Thursdays), and a smaller meeting after class on Tuesday. Individual members shall communicate to each other via chat.

8. Configuration management

SVN policy

1. The same SVN repository will be used for multiple projects, so every project should be placed in its own folder on the SVN.
2. Always write a short comment of the changes you are committing.
3. Every version committed to the SVN must compile.
4. Project's binary files shouldn't be committed, unless they are a part of some library that the project uses.
5. When resolving conflicts, be careful not to override other people's changes.

Files that should **NOT** be committed:
Visual Studio:

- All files from the **obj** and **bin** folders (*.exe, *.pdb, *.manifest, *.dll, …)
- Solution user options (***.suo**) files

Eclipse/NetBeans:

- All files from the bin folder (*.class, …)

9. Project plan

9.1 Time schedule

Id	Milestone Description	Responsible Dept./Initials	Finished week				Metr.	Rem.
			Plan	Forecast		Actual		
				Week	+/-			
M001	Requirements analysis & definition	DR, CF	41	41				
M002	Basic recording application	DR, NV	43	43				
M003	Basic player application	ŽR, NMK	43	43				
M004	Basic VLC streaming functionality	AT	43	43				
M005	System – Alpha version	Re + Pl + AT	44	44				
M006	Studio application - GUI	CF, DM	45	45				
M007	Studio application - complete	CF, DM, AT	47	47				
M008	System – Beta version	ALL	47	47				
M009	System – Release Candidate	ALL	51	51				
M010	Testing and Documentation	CF, DM, NV	1	1				

9.1.1 Remarks

Remark Id	Description

9.2 Activity plan

Activity	w 39	w 40	w 41	w 42	w 43	w 44	w 45	w 46	w 47	w 48	w 49	w 50	w 51	w 52	w 53	w 1	w 2
Project preparations	■																
Requirements analysis & definition	■	■	■														
Recorder app – video capture		■	■	■													
Recorder app – network streaming				■	■												
VLC server – setting up and config				■	■												
Player app – developing			■	■	■												
C# application - GUI				■	■	■											
C# application - core						■	■	■									
System integration								■	■					■	■		
Recorder app – additional features											■	■	■	■			
Player app – additional features											■	■	■	■			
C# application – feature upgrade											■	■	■	■			
Testing					■	■			■	■	■				■	■	■
Documentation	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
Presentations	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■

9.3 Financial Plan

Activity	Volume (days)	Cost	Rem.
Project preparations	5	1600$	
Requirements analysis & definition	7	2240$	
Recorder app – video capture	9	2880$	
Recorder app – network streaming	7	2240$	
VLC server – setting up and config	8	2560$	
Player app – developing	8	2560$	
C# application - GUI	9	2880$	
C# application - core	11	3520$	
System integration	13	4160$	
Recorder app – additional features	12	3840$	
Player app – additional features	12	3840$	
C# application – feature upgrade	12	3840$	
Testing	15	4800$	

Planned effort (man-days)	Man-day cost	Planned project cost (100%)
128	320$	40960$

9.3.1 Remarks

Remark Id	Description