
CUSTOMER DRIVEN PROJECT

Rock Concert Audience as a Screen

Project Report

Netlight AS

Agnethe Soraa, Tomas Dohnalek, Jan Bednarik, Milos Jovac
Project adviser: Anh Nguyen Duc

September 16, 2013

Abstract

The purpose of this document is to give an insight into the details of the planning, research, design and implementation of the task given in the course TDT4290 - Customer Driven Project. The project aims to give the students experience with a real project, and with a real customer. This gives the students an opportunity to combine both theory and practice. The customer for our project is Netlight AS.

Our project will be about researching and implementing image processing. Naturally this means we also have to solve problems regarding mapping of mocked units to locations as a function of time. The environment takes place at a rock concert, which means we also have to solve issues with timing and syncing between multiple independent units.

This is a proof-of-concept task. All the research done will be documented, and used to argue for and against the solutions. We will also argue for and against alternative solutions. Everything from the planning to the complete conclusion is described in this report. To be able to solve these problems we have to start by investigating relevant technologies, and how we can make this possible. The conclusion of this study allows us to create a system which showcases the real potential of our solution.

Preface

Contents

1	Introduction	4
1.1	General information	4
1.2	Terminology	4
1.3	Structure of report	4
1.4	Project and project name	4
1.5	Project purpose and concept	4
1.6	Project goal	4
1.7	Stakeholders	4
1.7.1	Customer	4
1.7.2	Customer contact	4
1.7.3	Development team	4
1.7.4	Advisor	4
1.8	Project background	4
2	Planning	5
2.1	Project plan	6
2.2	Methodology choice - Scrum	6
2.3	Organization	6
2.4	Risk Management	6
2.5	Quality Assurance	6
2.6	Measurement of project effects	6
2.7	Duration and workload	6
2.8	Gantt diagram	6
2.8.1	Description	6
2.8.2	Result schedule	6
2.8.3	Roles	6
2.8.4	Version Control	6
2.8.5	Textual documentation	6
3	Preliminary studies	7
3.1	Similar projects	7
3.2	Market investigation	7
3.3	Existing technologies and frameworks	7

3.4	Evaluation of alternative solutions	7
3.5	Outcome of research - Our decision	7
3.6	Constraints	7
3.7	Chosen development technologies??	7
3.8	Evaluation criteria	7
4	Requirements	8
4.1	Description/scope	8
4.2	Definitions/general terms	8
4.3	Business Requirements	8
4.3.1	Functional	8
4.3.2	Non-functional	9
4.4	Use cases	9
4.5	Summary	9
5	Testplan	10
5.1	Approach	10
5.2	Templates	10
5.3	Responsibilities	10
5.4	Test criteria	10
6	Software Architecture	11
6.1	Introduction	11
6.2	Selection of architectural viewpoints	11
6.3	Views	11
6.4	Tactics	11
6.5	Patterns	11
6.6	Data Storage	11
7	Tools and strategy	12
8	Sprint 0	13
8.1	Sprint planning	13
8.1.1	Sprint 0 User-stories	13
8.2	System Burndown	14
8.3	Architecture	15
8.4	Implementation	15
8.5	Testing	15
8.6	Occurring risks	15
8.7	Retrospective	15
8.7.1	Pros	15
8.7.2	Cons	15
8.8	Evaluation	15
9	Sprint 1	16
9.1	Sprint planning	16
9.1.1	Sprint1 User-stories	16

9.2	Architecture	17
9.3	Implementation	18
9.4	Testing	18
9.5	Occurring risks	18
9.6	Retrospective	18
9.6.1	Pros	18
9.6.2	Cons	18
9.7	Evaluation	18
10	Sprint 2	19
10.1	Sprint planning	19
10.1.1	User-stories	19
10.2	System Burndown	19
10.3	Architecture	19
10.4	Implementation	19
10.5	Testing	19
10.6	Occurring risks	19
10.7	Retrospective	19
10.7.1	Pros	19
10.7.2	Cons	19
10.8	Evaluation	19
11	Sprint 3	20
11.1	Sprint planning	20
11.1.1	User-stories	20
11.2	System Burndown	20
11.3	Architecture	20
11.4	Implementation	20
11.5	Testing	20
11.6	Occurring risks	20
11.7	Retrospective	20
11.7.1	Pros	20
11.7.2	Cons	20
11.8	Evaluation	20
12	Sprint 4	21
12.1	Sprint planning	21
12.1.1	User-stories	21
12.2	System Burndown	21
12.3	Architecture	21
12.4	Implementation	21
12.5	Testing	21
12.6	Occurring risks	21
12.7	Retrospective	21
12.7.1	Pros	21
12.7.2	Cons	21

12.8 Evaluation	21
13 Sprint 5	22
13.1 Sprint planning	22
13.1.1 User-stories	22
13.2 System Burndown	22
13.3 Architecture	22
13.4 Implementation	22
13.5 Testing	22
13.6 Occurring risks	22
13.7 Retrospective	22
13.7.1 Pros	22
13.7.2 Cons	22
13.8 Evaluation	22
14 Sprint 6	23
14.1 Sprint planning	23
14.1.1 User-stories	23
14.2 System Burndown	23
14.3 Architecture	23
14.4 Implementation	23
14.5 Testing	23
14.6 Occurring risks	23
14.7 Retrospective	23
14.7.1 Pros	23
14.7.2 Cons	23
14.8 Evaluation	23
15 Testing	24
15.1 Types	24
15.2 Unit testing	24
15.3 Integration	24
15.4 System testing	24
15.5 Usability	24
15.6 Acceptance	24
16 Evaluation	25
16.1 Group evaluation	26
16.1.1 Group dynamics	26
16.1.2 Role assignment	26
16.1.3 Risk evaluation	26
16.1.4 Customer and project task	26
16.1.5 Advisor	26
16.2 Project Evaluation	26
16.2.1 Planning	26
16.2.2 Preliminary Studies	26

16.2.3	Scrum	26
16.2.4	Meetings-Summary	26
16.2.5	Course feedback	26
16.2.6	Testing	26
16.2.7	Time usage	26
16.3	Technology evaluation	26
16.3.1	Skype	26
16.3.2	Github	26
16.3.3	Facebook	26
16.3.4	Testflight	26
16.3.5	Google documents	26
16.3.6	Latex	26
16.3.7	TargetProcess3	26
16.3.8	Technical issues	26
17	Conclusion	27
17.1	Introduction/Final product/description	27
17.2	Results	27
17.2.1	Functionalities	27
17.3	Evaluation criteria	27
17.4	Evaluation Results	27
17.5	Conclusion	27
17.6	Discussion	27
17.7	Further work	27
17.8	Reflection	27
17.9	Summary	27
18	References	28
19	Attachments	29
A	User Manual	30
B	Installation Guide	31
C	Glossary	32
D	XML Scheme?	33
E	Customer meetings	34
F	Group meetings	35
G	Supervisor meetings	36
H	Evaluation Questioner	37

1 | Introduction

1.1 General information

1.2 Terminology

1.3 Structure of report

1.4 Project and project name

1.5 Project purpose and concept

1.6 Project goal

1.7 Stakeholders

1.7.1 Customer

1.7.2 Customer contact

1.7.3 Development team

1.7.4 Advisor

1.8 Project background

2 | Planning

2.1 Project plan

2.2 Methodology choice - Scrum

2.3 Organization

2.4 Risk Management

2.5 Quality Assurance

2.6 Measurement of project effects

2.7 Duration and workload

2.8 Gantt diagram

2.8.1 Description

2.8.2 Result schedule

2.8.3 Roles

2.8.4 Version Control

2.8.5 Textual documentation

3 | Preliminary studies

3.1 Similar projects

3.2 Market investigation

3.3 Existing technologies and frameworks

3.4 Evaluation of alternative solutions

3.5 Outcome of research - Our decision

3.6 Constraints

3.7 Chosen development technologies??

3.8 Evaluation criteria

4 | Requirements

4.1 Description/scope

Since our customer namely specified the

4.2 Definitions/general terms

4.3 Business Requirements

4.3.1 Functional

1. Client behavior - Client must be able to receive commands from the server and display color according to the instructions.
2. More servers - An user can specify to which server (concert stage) he wants to connect.
3. Localization - Server must be able to detect mobile device's positions using image processing.
4. Core - Server should be able to display media through a few (a least 3) mobile phones' screens.
5. Media selection - An server user can choose which media should be played on screen made from mobile phones' screens.
6. Attendance - Number of connected devices to server must be displayed in server application.

4.3.2 Non-functional

Server-client architecture - Application must work as a server and client architecture. Platform - Audience application must work on at least one mobile platform. Deployment - Application must be deployed to relevant mobile application store. Scalability - The application must be scalable - it must work with different count of audience mobile phones. Generality - The application must be prepared for future using outside of rock concert domain. Delivery - Final product must be finished until 21th of November 2013 and presented to the committee and the customer.

4.4 Use cases

4.5 Summary

5 | Testplan

5.1 Approach

5.2 Templates

5.3 Responsibilities

5.4 Test criteria

6 | Software Architecture

6.1 Introduction

6.2 Selection of architectural viewpoints

6.3 Views

6.4 Tactics

6.5 Patterns

6.6 Data Storage

7 | Tools and strategy

8 | Sprint 0

8.1 Sprint planning

We have embraced Sprint 0 as a preliminary sprint, when we can set up all necessary collaboration tools, equipment, prepare templates for meetings and mainly to acquaint ourselves with Scrum methodology. The original plan was to finish sprint 0 on 8th of September, but we have decided to terminate it prematurely due to finishing sprint goals in shorter time than we had expected. Other reason for terminating the sprint was desire to start actually working on the product itself.

The actual user stories are listed in table 9.1. Since we started to use the software collaboration tool only during the sprint we did not manage to estimate the time needed to complete each story beforehand and thus the column **Est.** is left empty.

8.1.1 Sprint 0 User-stories

ID	Description	Est.	Sp.
259	I as a developer need to prepare \LaTeX template for minutes, project plan, weekly status report.		5
	Meeting minutes		2
	Project report		2
245	We as a team need to give a project and team name.		2
	Team name		1
	Product name		1
248	I as a developer need to agree on customer, advisor and internal meetings.		2
247	I as a developer need to agree on daily working hours.		1

243	I as a developer need to set up the video conferencing.	2
249	I as a developer need to add goals for Sprint 0.	4
250	I as a developer need to decide which collaboration technologies to use.	20
258	We as a team need to assign roles to team members.	1
258	I as a developer need to write a project plan.	90
258	I as a developer need to research the older reports.	30
258	I as a developer need to summarise the requirements.	4
SUM:		161

Table 8.1: User stories selected for Sprint 0.

8.2 System Burndown

Since we managed to establish the proper collaboration tool Target Process 3 only during the sprint the software was not able to generate relevant burndown chart. We at least tried to estimate how much time we spent working on each of the user stories listed in table

8.3 Architecture

8.4 Implementation

8.5 Testing

8.6 Occurring risks

8.7 Retrospective

8.7.1 Pros

8.7.2 Cons

8.8 Evaluation

9 | Sprint 1

9.1 Sprint planning

After assembling all the tools in Sprint0, we decided to start with the implementation of core modules. As our understanding of task improved, we were able to come up with user stories from the perspective of user, customer, developer and student. All user-stories were given to the customer so they can be prioritized. All but user-stories concerning our student obligations, like writing project plan, minutes, meetings with supervisor and attending lectures. Those were mandatory and already added as user-stories of sprint1. On Monday 02.09.2013. we had the meeting with a customer where we estimated time we need for every user story. The result of that meeting was the list of the rest of the user-stories for sprint1. All user stories for finishing our first prototype were on the sprint1 list so we also agreed date for presentation and showing the running demo - Thursday 12.09.2013. After that ,at a group meeting, we decoupled user-stories into tasks and we were ready to start with the implementation of client-server core module.

9.1.1 Sprint1 User-stories

ID	Description	Est.	Sp.
353	I as a developer need to make client receive commands from the server.	4	4
345	Customer meeting.	6	6
344	Team building.	7	9
314	I as a developer need to put "Hello World" project to gitHub and pull it to every group member's local storage Create folder on gitHub account named "source". Install ADT and Eclipse to our local computers. Create new Android Project and push it to gitHub.	18	4.7

267	As a user I want to easily download the app from testflight. Set up testflight. integrate testflight SDK.	5	5
312	I as a developer need to make server to be able to listen for the clients. Research about server sockets. Implement server listener. Create the moc client. Connect with mock client.	30	22
335	The server sends one command to one client.	4	4
336	The client receives one command.	2	2
334	The client "plays" one command (white light 10 seconds).	4.5	4.5
327	As a students we need to attend a meeting with our supervisor. Attend meeting with supervisor week1 (06.09.2013). Attend meeting with supervisor week2 (13.09.2013).	16	16
321	I as a student need to participate to lectures about team dynamics this week. Course of group dynamics Thu. Summary of course and exchange learned.	32	25
290	As a user I want to see the number of connected devices.	0.5	0.5
341	Integrate TestFlight into application.	15	3
343	As developer I have to work on Project Plan.	12	12
313	I as a developer need to establish basic communication protocol between client & server.	4	4
262	I as a developer need to research TestFlightApp. Figure out whether to use HockeyApp or TestFlight Research TestFlight	6	2.5
SUM:		164	?

Table 9.1: User stories selected for Sprint 0.

9.2 Architecture

Choosing client-server architecture was very intuitive to do. Our project has user application that depends on commands for what to play, on one side, and application that is responsible of detecting and sending commands to that users on the other. Every application(user) have to be either one or another.

Write about Android NSD, create class diagram,

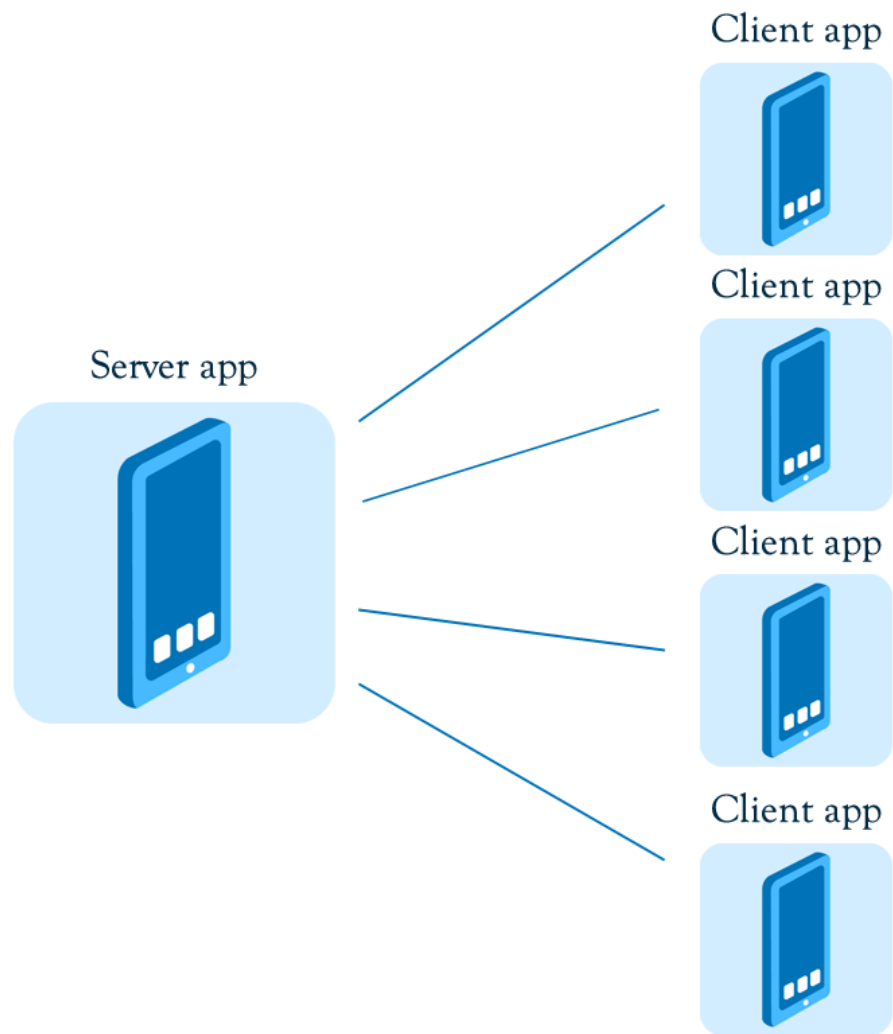


Figure 9.1: Sprint1 Architecture

9.3 Implementation

9.4 Testing

9.5 Occurring risks

18

9.6 Retrospective

9.6.1 Pros

9.6.2 Cons

10 | Sprint 2

10.1 Sprint planning

10.1.1 User-stories

10.2 System Burndown

10.3 Architecture

10.4 Implementation

10.5 Testing

10.6 Occurring risks

10.7 Retrospective

10.7.1 Pros

10.7.2 Cons

10.8 Evaluation

11 | Sprint 3

11.1 Sprint planning

11.1.1 User-stories

11.2 System Burndown

11.3 Architecture

11.4 Implementation

11.5 Testing

11.6 Occurring risks

11.7 Retrospective

11.7.1 Pros

11.7.2 Cons

11.8 Evaluation

12 | Sprint 4

12.1 Sprint planning

12.1.1 User-stories

12.2 System Burndown

12.3 Architecture

12.4 Implementation

12.5 Testing

12.6 Occurring risks

12.7 Retrospective

12.7.1 Pros

12.7.2 Cons

12.8 Evaluation

13 | Sprint 5

13.1 Sprint planning

13.1.1 User-stories

13.2 System Burndown

13.3 Architecture

13.4 Implementation

13.5 Testing

13.6 Occurring risks

13.7 Retrospective

13.7.1 Pros

13.7.2 Cons

13.8 Evaluation

14 | Sprint 6

14.1 Sprint planning

14.1.1 User-stories

14.2 System Burndown

14.3 Architecture

14.4 Implementation

14.5 Testing

14.6 Occurring risks

14.7 Retrospective

14.7.1 Pros

14.7.2 Cons

14.8 Evaluation

15 | Testing

15.1 Types

15.2 Unit testing

15.3 Integration

15.4 System testing

15.5 Usability

15.6 Acceptance

16 | Evaluation

16.1 Group evaluation

16.1.1 Group dynamics

16.1.2 Role assignment

16.1.3 Risk evaluation

16.1.4 Customer and project task

16.1.5 Advisor

16.2 Project Evaluation

16.2.1 Planning

16.2.2 Preliminary Studies

16.2.3 Scrum

16.2.4 Meetings-Summary

16.2.5 Course feedback

16.2.6 Testing

16.2.7 Time usage

16.3 Technology evaluation

16.3.1 Skype

16.3.2 Github

16.3.3 Facebook

16.3.4 Testflight

17 | Conclusion

17.1 Introduction/Final product/description

17.2 Results

17.2.1 Functionalities

17.3 Evaluation criteria

17.4 Evaluation Results

17.5 Conclusion

17.6 Discussion

17.7 Further work

17.8 Reflection

17.9 Summary

18 | References

19 | Attachments

A | User Manual

B | Installation Guide

C | Glossary

D | XML Scheme?

E | Customer meetings

F | Group meetings

G | Supervisor meetings

H | Evaluation Questioner