Middle East Technical University
Department of Computer Engineering

Project Proposal
for
“CL@SS++”

profIT

1250125 - İsmail ÇETİN
1250349 - Yavuz GÜRCAN
1250356 - Recep GÜRLEK
1250661 - Hakan ÖZTÜRK

Fall 2004-2005

Table of Contents

1 Introduction

With the developments in technology and science, new techniques are started to be used in various areas instead of the classical methods. These have also affected a key field which is education from different aspects. Recently a trend towards computer-aided teaching techniques has started. In this respect, instructors use extensively slide shows as well as the course web page for reference materials. Such materials are reusable and prepare a basis for the future online educational resources. As these developments are in a continuous progress, the next phase will be online education accessible from any internet connected device.

Since this progress is still in its early stages, it has still many insufficiencies. Therefore, we aim to develop a tool for facilitating synchronous distance learning with a wide range of multimedia facilities.

2 Project Scope

2.1 Purpose

The aim of the Project is to develop a tool for synchronous distance learning with multimedia facilities. The CL@SS++ Project will be a web-based graphical user interface (GUI) that facilitates education over the internet. This tool will be of use both to the instructors as well as to the students. A student with a basic computer experience can easily follow his/her lectures via internet. Similarly, the instructors can give their lectures over the internet and provide the students with additional instructional materials without difficulty.

2.2 Features

We divided the features according to the actors. These features are the result of the brainstorming sessions in the meetings (see Appendix B: Initial Meeting – Minutes) and investigation of the existing systems. The features may be enhanced or reduced in the analysis and design phases. The related features are listed in the next sections for the instructor, student and administrator separately.

2.2.1 Features for the Instructor

The instructor…

- can create/delete courses
- will have full access to the course materials
- can broadcast the video taken live from his webcam
- can make his/her presentations visible to the students
- can display his/her desktop simultaneously
- may prefer to write on a Paint-like program with the mouse, instead of writing on a blackboard

- can follow the attendance of the student(s), taken automatically by the program
- can track the actions taken by the students on the platform related with the course
    - When visited
    - How long visited
    - What downloaded
    - What displayed
    - and other related information stored
- can prepare exams and make them available
    - the tool can provide a exam preparation interface that facilitates multiple-choice and classical question preparation, and reflect the result to the student as a form
    - s/he may make a pop-quiz
        - prepared beforehand
        - prepared immediately during the lecture
    - s/he may make an examination and set a date interval to be visible to the student
- can give a homework and the tool will handle the submission issues
- may take questions during the lecture, as well as other times
- if the students have a webcam, may see the students as well
- if the students have a microphone, may hear the students questions
- may take the snapshot of the current image, and make it available to all students
    - This will enable the access to past information. For example, the instructor may use the blackboard, and before erasing the whole blackboard, s/he can take a snapshot so that the students can follow the previous state of the blackboard during the lecture.
- can prepare an outline of the lecture
    - During the lecture, the instructor will press on the related item of the outline. This will enable the students to trace back to the beginning of the discussion of the item within the video. For example, assume that the topics are “Introduction”, “teamwork”, and “innovative thinking”. When the instructor marks every item on the beginning of that item, the student will be able to start the video from that point.
    - The discussed items will be marked by the system so that the students can track what has been studied so far.
- can prepare a “related links” section
- can prepare answers to common questions
- can upload tests, presentations and videos related with the course to be available to the students
- can update and delete documents
- can e-mail all students or a set of students
- can mark a video if a mistake was made
    - For example, when the student downloads a video with a mistake in it, s/he will be informed that “in time interval *x* by mistake *y* is said but it should be *z*”.
- should be able to see whether he has a proper connection

- should be able to read and post messages in the forum

2.2.2 Features for the Student

The student…

- can see the broadcasted video of the lecture
- may trace back in the stream (even during the live session)
    - may return to the latest live stream thereafter
- can see an outline (if provided by the instructor), and see the progress so far
- can trace the video by clicking on the outline items, and watch the related part of the video
- can listen to the broadcasted lecture
- has access to the documents, links, questions provided by the instructor
- can watch the lecture even if offline
- can download a previous lecture
- can ask questions during the lecture (if the instructor allows), or in the forum whenever s/he wants
- can take the snapshot of the video (for personal usage)
- may connect with a webcam
- may connect with a microphone
    - and may ask questions with the microphone
- may be offered with alternative connection qualities
- may see his/her homework, and submit it
- can send an e-mail to the instructor
- can check his connection
- can follow his progress

2.2.3 Features for the Administrator

There should also be an administrator with a full power over the system. Some basic features for the administrator can be listed as follows:
The administrator…

- can add/remove users
- can add/remove classes/courses
- can assign users to classes/courses
- can update user/course/class information

3 Software Methodology

Nowadays, it is not only the implementation of a project that is important, but also the analysis and design phases are crucial in maintaining Projects. Therefore, the industry is widely applying accepted methodologies. The current methodology is Object-Oriented modeling. In this respect, many Projects are built with the Unified Modeling Language (UML) [1].

We will also make use of UML, as it will be useful for the team members to practice and apply UML in the Project and they will benefit from it after graduation. In addition, Rational Rose Enterprise Edition [2], which is the most widely used CASE tool in the industry, will be used for modeling the project.

We have also inspected other CASE tools. However, no other CASE tool seems to facilitate the process as much as Rational Rose. One of these is Poseidon for UML by Gentleware [3]. It seems that the Community Edition of this tool is quite restricted and the Professional Edition is not as user-friendly as Rational Rose. Moreover, the team members prefer to use and become familiar with a CASE tool the industry makes use of.

Another issue about the suitability of the methodology to the project is that UML seems to be more a tool to develop classes for Object-Oriented Projects. However, we are currently studying and researching to enable UML within the Project. For example, in "Modeling Web Applications Architecture with UML" [4]:

> "*…When attempting to model web applications with UML, it becomes apparent that some of its components don't fit nicely into standard UML modeling elements. In order to stick with one modeling notation for the entire system (web components and traditional middle tier components), UML must be extended. This paper presents an extension to the UML, using its formal extension mechanism. The extension is designed so that web-specific components can be integrated with the rest of the system's model, and to exhibit the proper level of abstraction and detail suitable for designers, implementers, and architects of web applications*."

Other similar resources for Web Application Development with UML we have identified are [5, 6]. We will research and study this methodology before and during the analysis phase.

Similarly, we are in progress of studying Graphical User Interface Design. This process has to do with software Engineering methodologies together with Image Processing and Psychological aspects. Another important aspect is network broadcasting. Obviously, this concept should be studied in detail, too.

4 Project Resources

The Project resources are the team itself, hardware and software to be used. The following sections give detailed information about the resources.

4.1 Staff Organization

4.1.1 Team structure

Since the team is small, the team structure will be "Controlled Decentralized" (CD) with one Team Leader. Problem solving is still a group activity. Moreover, the team is open to new ideas. With this model coordination of the team will be easier and the work efficiency will maximize. The following section lists the members with project related information.

4.1.2 Team Members

Name	Yavuz Gürcan
Skills	Java (J2SE, J2EE), JDBC, SQL, JXTA, XML, DTD, HTML, Java Servlet Technology, JavaScript, C/C++, PHP, MySQL, Apache Tomcat, Axis, SOAP, Eclipse, Poseidon, MS Project, Rational Rose, MS Office, Matlab, Poseidon for UML, Dreamweaver
E-mail	yavuz@srdc.metu.edu.tr e1250349@ceng.metu.edu.tr
Phone	0 505 5116886
Roles in Project	Leader, Analyst, Designer, Implementer, Tester
Roles in the Meetings	Initiator, Chair, Timekeeper, Gate Keeper

Name	Hakan Öztürk
Skills	C/C++, JavaScript, ASP, HTML, Matlab, FrontPage, SQL, MS Project, MS Office
E-mail	hakannazim1@gmail.com e1250661@ceng.metu.edu.tr
Phone	0 505 5041057 0 535 3832012
Roles in Project	Analyst, Designer, Implementer, Tester
Roles in the Meetings	Initiator, Recorder

Name	Recep Gürlek
Skills	C/C++, Java, SQL, JSP, XML, XSLT, HTML, CSS, JavaScript, Matlab, Apache Tomcat, JBoss, JDBC, MS Project, MS Office, PHP, MySQL
E-mail	recepgurlek@gmail.com e1250356@ceng.metu.edu.tr
Phone	0 542 6744954
Roles in Project	Analyst, Designer, Implementer, Tester
Roles in the Meetings	Initiator, Summarizer, Skeptic

Name	İsmail Çetin
Skills	C/C++, Java, SQL, JSP, XML, XSLT, HTML, CSS, JavaScript, Matlab, Apache Tomcat, JBoss, JDBC, MS Project, MS Office, Dreamweaver, PHP, MySQL
E-mail	ismail.cetin@gmail.com e1250125@ceng.metu.edu.tr
Phone	0 536 9658273
Roles in Project	Analyst, Designer, Implementer, Tester
Roles in the Meetings	Initiator, Optimist

4.2 Hardware Requirements

4.2.1 Development

- 4 computers with
    - P4 CPU or compatible
    - 512 MB RAM
    - Internet connection
    - Ethernet card
    - 1 GB free disc space

4.2.2 Client-side

4.2.2.1 Instructor

- A computer with
    - P4 CPU or compatible
    - 512 MB RAM
    - Internet connection
    - Ethernet card
    - 1 GB free disc space
- Web cam
- Microphone
- High bandwidth data connection cables
- Speaker
- Sound Card

4.2.2.2 Student

- A computer with
    - P4 CPU or compatible
    - 256 MB RAM
    - Internet connection
    - Ethernet card
    - 2 GB free disc space
- Web cam (Optional)
- Microphone (Optional)
- High bandwidth data connection cables
- Speaker
- Sound Card

4.2.3 Server-side

- A computer with
    - P4 CPU or compatible
    - 2 GB RAM
    - Internet connection
    - Ethernet card
    - 80 GB Hard Disk
    - High bandwidth data connection cables

4.3 Software Requirements

4.3.1 Development

- Programming Languages and Development Environments
- Web Server
- Rational Rose as CASE Tool
- MS Project
- Internet Explorer 6.0+ or Compatible
- Windows XP or Windows 2003 Server
- Oracle Database
- Macromedia Flash and Dreamweaver
- Windows Media Player or Multimedia Component

4.3.2 Client-side

4.3.2.1 Instructor

- Internet Explorer 6.0+ or Compatible Web Browser
    - Flash Plug-in
    - Java Runtime Environment
- Windows XP
- Windows Media Player or Multimedia Component
- Video Capturing Software

4.3.2.2 Student

- Internet Explorer 6.0+ or Compatible Web Browser
    - Flash Plug-in
    - Java Runtime Environment
- Windows XP
- Windows Media Player or Multimedia Component
- Video Capturing Software (Optional)

4.3.3 Server-side

- Oracle Database
- Web Server
- Windows 2003 Server

5 Project Schedule

This section presents an overview of project tasks and the output of the project scheduling tool.

5.1 Project Milestones

The Project Milestones are as follows:

Proposal Report	10.10.2004
Requirements Analysis Report	02.11.2004
Initial Design Report	03.12.2004
Detailed Design Report	03.01.2004
Prototype Demo	18.01.2004
Final Product	27.05.2004

5.2 Tasks with Assignments

The following table lists all the tasks identified together with their start and finish dates, duration, and resources assigned to them. The progress of each item can be followed from the "% Work Complete" column. This table is created taking into consideration a 7-day working week, excluding national holidays. Team means that the work is done by the whole team. Additionally, the resources have the "*job:resource_name*" structure.

ID	Task Name	Work	Duration	Start	Finish	% Work Complete	Resource_Names
1	**Proposal**	88 hrs	10 days	Fri 10/1/04	Sun 10/10/04	100%	
2	Determine project scope	48 hrs	6 days	Fri 10/1/04	Fri 10/8/04	100%	Team
3	Define preliminary resources	16 hrs	2 days	Wed 10/6/04	Fri 10/8/04	100%	Team
4	Obtain core resources	16 hrs	2 days	Thu 10/7/04	Fri 10/8/04	100%	Team
5	Finalize Project Proposal Report	8 hrs	1 day	Sat 10/9/04	Sun 10/10/04	100%	Team

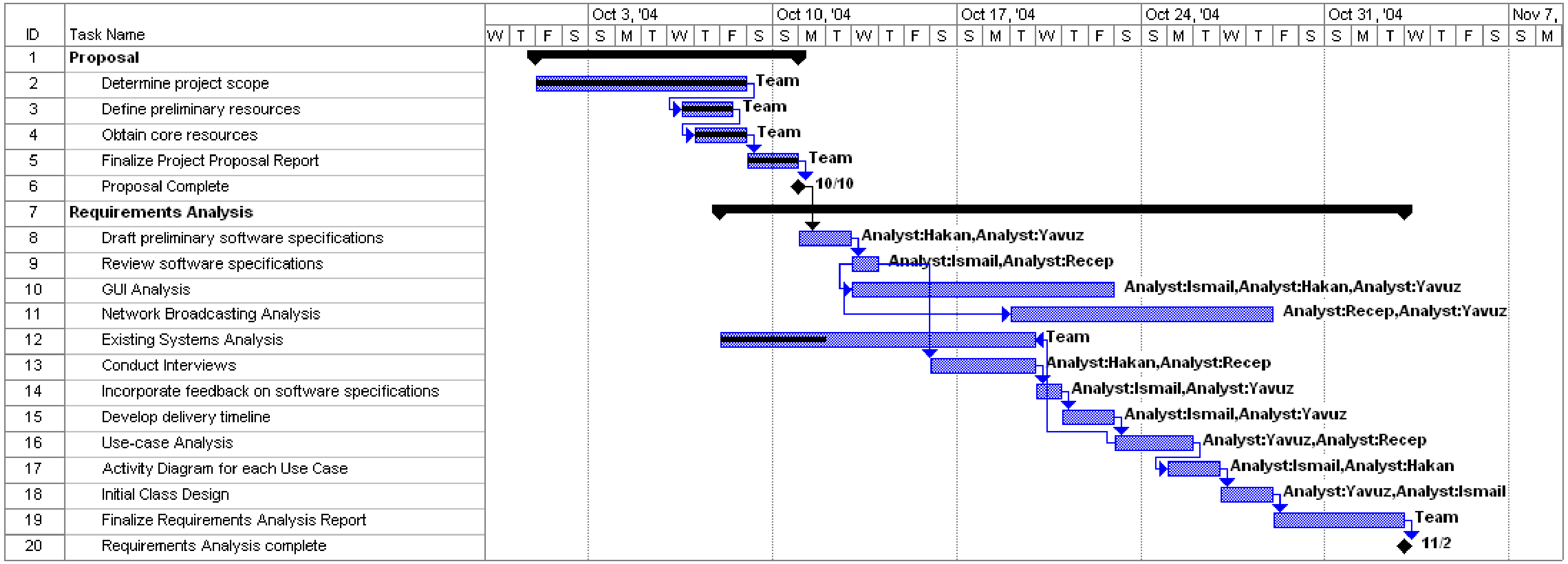
6	Proposal Complete	0 hrs	0 days	Sun 10/10/04	Sun 10/10/04	100%	
7	**Requirements Analysis**	776 hrs	26 days	Fri 10/8/04	Tue 11/2/04	2%	
8	Draft preliminary software specifications	32 hrs	2 days	Mon 10/11/04	Tue 10/12/04	0%	Analyst:Hakan, Analyst:Yavuz
9	Review software specifications	16 hrs	1 day	Wed 10/13/04	Wed 10/13/04	0%	Analyst:Ismail, Analyst:Recep
10	GUI Analysis	240 hrs	10 days	Wed 10/13/04	Fri 10/22/04	0%	Analyst:Ismail, Analyst:Hakan, Analyst:Yavuz
11	Network Broadcasting Analysis	160 hrs	10 days	Tue 10/19/04	Thu 10/28/04	0%	Analyst:Recep, Analyst:Yavuz
12	Existing Systems Analysis	80 hrs	10 days	Fri 10/8/04	Wed 10/20/04	20%	Team
13	Conduct Interviews	64 hrs	4 days	Sat 10/16/04	Tue 10/19/04	0%	Analyst:Hakan, Analyst:Recep
14	Incorporate feedback on software specifications	16 hrs	1 day	Wed 10/20/04	Wed 10/20/04	0%	Analyst:Ismail, Analyst:Yavuz
15	Develop delivery timeline	32 hrs	2 days	Thu 10/21/04	Fri 10/22/04	0%	Analyst:Ismail, Analyst:Yavuz
16	Use-case Analysis	48 hrs	3 days	Sat 10/23/04	Mon 10/25/04	0%	Analyst:Yavuz, Analyst:Recep
17	Activity Diagram for each Use Case	32 hrs	2 days	Mon 10/25/04	Tue 10/26/04	0%	Analyst:Ismail, Analyst:Hakan
18	Initial Class Design	32 hrs	2 days	Wed 10/27/04	Thu 10/28/04	0%	Analyst:Yavuz, Analyst:Ismail
19	Finalize Requirements Analysis Report	24 hrs	3 days	Fri 10/29/04	Tue 11/2/04	0%	Team
20	Requirements	0 hrs	0 days	Tue 11/2/04	Tue 11/2/04	0%	

	Analysis complete						
21	**Design**	1,416 hrs	58 days	Wed 11/3/04	Mon 1/3/05	0%	
22	Review preliminary software specifications	24 hrs	3 days	Wed 11/3/04	Fri 11/5/04	0%	Team
23	Develop functional specifications	56 hrs	7 days	Sat 11/6/04	Wed 11/17/04	0%	Team
24	Review functional specifications	32 hrs	4 days	Thu 11/18/04	Tue 11/23/04	0%	Team
25	GUI Design	720 hrs	30 days	Mon 11/29/04	Tue 12/28/04	0%	Designer:Yavuz, Designer:Ismail, Designer:Hakan
26	Initial Design Report	48 hrs	6 days	Fri 11/26/04	Fri 12/3/04	0%	Team
27	Finalize Initial Design Report	0 hrs	0 days	Fri 12/3/04	Fri 12/3/04	0%	
28	**Object-Oriented Design**	480 hrs	30 days	Wed 11/24/04	Thu 12/23/04	0%	
29	Class Diagram Creation	64 hrs	4 days	Wed 11/24/04	Sat 11/27/04	0%	Designer:Recep, Designer:Yavuz
30	Sequence Diagram for each Use Case	64 hrs	4 days	Sun 11/28/04	Wed 12/1/04	0%	Designer:Hakan, Designer:Ismail
31	Class Diagram Update	32 hrs	2 days	Thu 12/2/04	Fri 12/3/04	0%	Designer:Ismail, Designer:Recep
32	Statechart for Major Dynamic Classes	96 hrs	6 days	Sat 12/4/04	Thu 12/9/04	0%	Designer:Yavuz, Designer:Hakan
33	Collaboration Diagram for each Use Case	96 hrs	6 days	Fri 12/10/04	Wed 12/15/04	0%	Designer:Yavuz, Designer:Recep
34	Finalizing Class Diagram	128 hrs	8 days	Thu 12/16/04	Thu 12/23/04	0%	Designer:Yavuz, Designer:Hakan
35	Finalize Design Report	56 hrs	7 days	Fri 12/24/04	Mon 1/3/05	0%	Team

36	Design Complete	0 hrs	0 days	Mon 1/3/05	Mon 1/3/05	0%	
37	**Prototype**	360 hrs	27 days	Mon 12/13/04	Tue 1/18/05	0%	
38	Review functional specifications	16 hrs	2 days	Mon 12/13/04	Tue 12/14/04	0%	Team
39	Assign development staff	24 hrs	3 days	Wed 12/15/04	Fri 12/17/04	0%	Developer:Yavuz
40	Develop code	136 hrs	17 days	Mon 12/20/04	Tue 1/11/05	0%	Team
41	Developer testing (primary debugging)	80 hrs	5 days	Fri 1/7/05	Thu 1/13/05	0%	Team[200%]
42	Update Code	104 hrs	6.5 days	Mon 1/10/05	Tue 1/18/05	0%	Team[200%]
43	Prototype complete	0 hrs	0 days	Tue 1/18/05	Tue 1/18/05	0%	
44	Implementation	560 hrs	70 days	Mon 2/21/05	Fri 5/27/05	0%	Team

5.3 Gantt chart

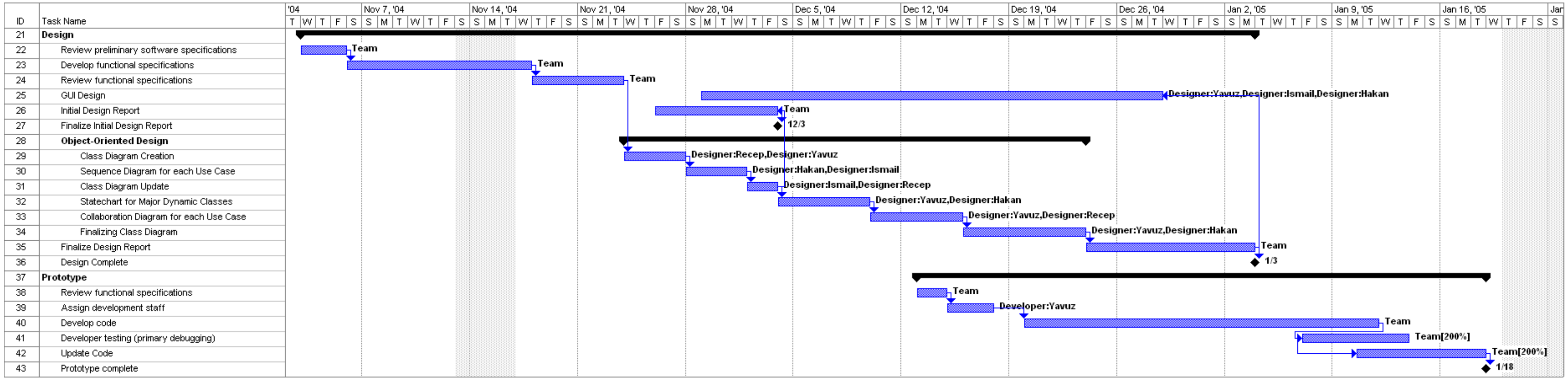
The following figures are the Gantt chart for the whole project. Figure 1 corresponds to the part from the beginning till the end of the Requirements Analysis Phase. Figure 2 covers the design and prototype implementation phases. Finally, Figure 3 although not detailed, represents the implementation phase.



ID	Task Name
1	**Proposal**
2	Determine project scope
3	Define preliminary resources
4	Obtain core resources
5	Finalize Project Proposal Report
6	Proposal Complete
7	**Requirements Analysis**
8	Draft preliminary software specifications
9	Review software specifications
10	GUI Analysis
11	Network Broadcasting Analysis
12	Existing Systems Analysis
13	Conduct Interviews
14	Incorporate feedback on software specifications
15	Develop delivery timeline
16	Use-case Analysis
17	Activity Diagram for each Use Case
18	Initial Class Design
19	Finalize Requirements Analysis Report
20	Requirements Analysis complete

Oct 3, '04
Oct 10, '04
Oct 17, '04
Oct 24, '04
Oct 31, '04
Nov 7,
W T F S S M T W T F S S M T W T F S S M T W T F S S M T W T F S S M T W T F S S M
Team
Team
Team
Team
10/10
Analyst:Hakan,Analyst:Yavuz
Analyst:Ismail,Analyst:Recep
Analyst:Ismail,Analyst:Hakan,Analyst:Yavuz
Analyst:Recep,Analyst:Yavuz
Team
Analyst:Hakan,Analyst:Recep
Analyst:Ismail,Analyst:Yavuz
Analyst:Ismail,Analyst:Yavuz
Analyst:Yavuz,Analyst:Recep
Analyst:Ismail,Analyst:Hakan
Analyst:Yavuz,Analyst:Ismail
Team
11/2

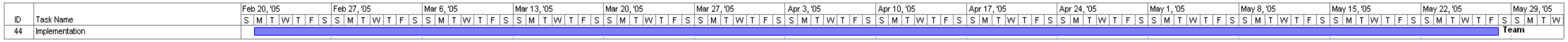
Figure 1 Gantt chart - Part 1



ID	Task Name
21	**Design**
22	Review preliminary software specifications
23	Develop functional specifications
24	Review functional specifications
25	GUI Design
26	Initial Design Report
27	Finalize Initial Design Report
28	**Object-Oriented Design**
29	Class Diagram Creation
30	Sequence Diagram for each Use Case
31	Class Diagram Update
32	Statechart for Major Dynamic Classes
33	Collaboration Diagram for each Use Case
34	Finalizing Class Diagram
35	Finalize Design Report
36	Design Complete
37	**Prototype**
38	Review functional specifications
39	Assign development staff
40	Develop code
41	Developer testing (primary debugging)
42	Update Code
43	Prototype complete

'04
Nov 7, '04
Nov 14, '04
Nov 21, '04
Nov 28, '04
Dec 5, '04
Dec 12, '04
Dec 19, '04
Dec 26, '04
Jan 2, '05
Jan 9, '05
Jan 16, '05
Jar
Team
Team
Team
Designer:Yavuz,Designer:Ismail,Designer:Hakan
Team
12/3
Designer:Recep,Designer:Yavuz
Designer:Hakan,Designer:Ismail
Designer:Ismail,Designer:Recep
Designer:Yavuz,Designer:Hakan
Designer:Yavuz,Designer:Recep
Designer:Yavuz,Designer:Hakan
Team
1/3
Team
Developer:Yavuz
Team
Team[200%]
Team[200%]
1/18

Figure 2 Gantt chart - Part 2



ID	Task Name
44	Implementation

Figure 3 Gantt chart - Part3

6 Conclusion

In this report we made a proposal for the CL@SS++ Project. We tried to formulate the scope and give initial features of the proposed system. Moreover, we identified the hardware and software requirements and introduced our team in the "Project Resources" section. We outlined the methodology we plan to use throughout the project and indicated our initial attempts in the "Software Methodology" section. In addition we developed an initial schedule for the project and assigned responsible team members to each task. We conducted several meetings and added the Agenda (see Appendix A: Initial Meeting – Agenda) and Meeting Minutes (see Appendix B: Initial Meeting – Minutes) as Appendices to the end of the report. Obviously, there is still much to do in the Project.

7 References

[1] The Unified Modelling Language (UML) Resource Page, http://www.uml.org/
[2] Rational Rose, http://www-306.ibm.com/software/rational/
[3] Poseidon for UML, Gentleware Homepage, http://www.gentleware.com/
[4] Jim Conallen, "Modeling Web Application Architectures with UML", June 1999, October 1999 (volume 42, number 10) issue of Communications of the ACM., http://www3.software.ibm.com/ibmdl/pub/software/rational/web/whitepapers/2003/webapps.pdf
[5] Jim Conallen, "Modeling Web Applications with UML", http://www.conallen.org/whitepapers/webapps/ModelingWebApplications.htm
[6] "Chapter 19: Web Modeling", "Mastering UML with Rational Rose 2002" by Wendy Boggs and Michael Boggs, Sybex Inc., http://www.srdc.metu.edu.tr/~yavuz/Ceng490/resources/Sybex%20-%20Mastering%20UML%20With%20Rational%20Rose%202002.pdf

8 Appendix

8.1 Appendix A: Initial Meeting – Agenda

profIT 07.10.2004 Meeting - Agenda

Date:	07.10.2004
Time:	19.30
Location:	Department of Computer Engineering
Attendees Expected:	- Yavuz Gurcan - Hakan Ozturk - Ismail Cetin - Recep Gurlek
End Time:	(Around) 21.00

Issues to be discussed (in order):

- Setting Ground Rules
- Setting Communication means and rules
- Setting up available times table
- Restating Project Goals
- Deciding on Project Title
- Talking about features of the Project
- Talking about Roles and Responsibilities
- Developing Project Proposal Outline
- Things to do next and sharing the work

For the meeting:

- Be prepared!
- If you feel there are other items that should be discussed, please inform me. I will change the list accordingly. If you think everything is OK, I will still appreciate getting a response…
- Think about each item, and have some input please. Especially:
    - Think first about the project and new features without analyzing "existing tools". After the meeting we will also analyze existing tools and enhance the features. I believe this way it would be better.
    - Think about roles and responsibilities you can take (or others should take).
    - Check some Project Proposals, Templates or Checklists.
    - Have your available and preferred meeting times ready (Both for the assistant and project group meetings).
- Be there ;)

Yavuz G.
profIT

8.2 Appendix B: Initial Meeting – Minutes

profIT 07.10.2004 Meeting - Minutes

Date:	07.10.2004
Time:	19.30
Location:	Department of Computer Engineering – Study Room
Attendees:	

- Yavuz Gurcan – Leader, Meeting Chair, Initiator
- Hakan Ozturk - Recorder, Initiator
- Ismail Cetin - Initiator
- Recep Gurlek - Initiator

End Time: 20.50

Issues discussed:

- Setting Ground Rules
- Setting Communication means and rules
- Setting up available times table
- Restating Project Goals
- Deciding on Project Title
- Talking about features of the Project
- Talking about Roles and Responsibilities
- Developing Project Proposal Outline
- Things to do next and sharing the work

Minutes:

- Ground Rules are discussed. In this respect the team members agreed on the following rules:
    - The team will meet every week, 2 days before the meeting with the assistant
        - In case the meeting can not be held, the next day a meeting will be held
    - Everyone should come prepared to the meeting
    - Every deliverable should be read and reviewed by each member
    - Everybody agrees on the rules. If required new rules can be added.
- The members decided that cellular phone, sms, and e-mails are the communication means.
    - The preferred e-mail addresses of each member are as follows:
        - hakannazim1@yahoo.com, Hakan Ozturk
        - recepgurlek@gmail.com, Recep Gurlek
        - ismail.cetin@gmail.com, Ismail Cetin
        - yavuz@srdc.metu.edu.tr, Yavuz Gurcan
    - The members have also exchanged their phone numbers
    - When e-mails are received, everybody should reply, so that, the sender at least knows that his mail arrived and was read.
    - When an SMS is received, the sender will be ringed
- The team members combined their weekly schedules
    - The available time slots to meet with the assistant are identified and numbered according to preference. The members decided on the following time slots:
        1. Monday 10.40-11.30
        2. Tuesday 16.40-17.30
        3. Wednesday 9.40-10.30
        4. Friday 10.40-11.30
        5. Friday 13.40-17.30
    - The other time slots are unavailable due to several reasons. Upon request the assistant can be convinced with a detailed schedule.
    - For the weekly team meetings:
        - Recep requested that when meetings are held on weekends, they better be during the day, not on evenings.
        - Hakan said that he has football games on Wednesdays 21.00-22.00, and Thursdays 22.00-23.00
- The project goals are restated and everybody agreed on the goals. The team members had in general the same ideas about the project. The following items are identified:
    - A video capture device is required
    - User should be able to load lectures
    - Users can ask questions during the lecture (if allowed by instructor) and/or after
    - As the project deals with multimedia, network broadcasting and GUI, GUIs should be researched
    - A database should be set up and accessible
    - A tool to prepare videos may be required, in case the instructor wishes to save his lecture beforehand
- The team members brainstormed for a Project title. Several ideas were proposed, but none was acceptable. Therefore, everybody is supposed to think of a Project Title till Saturday.
- The team members were supposed to think about features of the project. A brainstorming session for the features started. The following items were identified:

  - Besides broadcasting other helper visuals should be displayed to the client:
    - Presentation
    - PC Desktop
    - Video stream from Camera…
  - Both the instructor and students should be able to take the snapshot of the moment. When instructors take a snapshot it will be available to all clients, students can take separate snapshots.
  - The instructor can give input from the keyboard
  - The students may interrupt the instructor
  - The instructor can make pop-quizzes or exams online. For this a tool can be developed to provide the instructor an environment to prepare multiple choice or classic questions and answer sheets. These can be given immediately (pop-quiz) or a time can be adjusted.
  - The instructor can add a FAQ section, links, documents, tests, presentations and videos related with the course
  - Automatic attendance can be taken. Besides, information about the students like:
    - When visited
    - How long visited
    - What downloaded
    - What displayed
    - And other related information can be stored
  - According to the users' communication quality, a video broadcasting quality should be offered
  - Homework can be given on the page. The students can also submit their homework from the page
  - There exist a technology with electronic board and special pencils. Such technologies are used in similar projects. However, as we don't have such a technology available we will not concentrate on this.
  - The instructor may choose to write on a Paint-like environment instead of a blackboard. With the mouse it is difficult, but there are pen-mouses?
  - The student may trace back in the video, as there is also a local copy of the previous parts of the video. The user may also get back to live stream.
  - The instructor may not use a blackboard, and prefers paint. The desktop should be reflected.
  - The instructor should be able to update and delete documents
  - There can be an e-mail warning and information mechanism
  - In case a mistake is made in a video, the mistake can be corrected. For example, when the student watches a video from the archive he will see a warning that states "in the *x* part of the video *y* is said, but it should be *z!*"
  - The video can be marked with an outline. The instructor prepares an outline and marks the related sections (during broadcasting). The student may go to the related part by clicking on the outline item.
  - The discussed topics in the outline may be marked with another color.
  - A check whether the video can be displayed by the student or not. Moreover, another check, whether a stream is available from the instructor or not may be needed.
  - The instructor may see the students as well, when they have a cam.
  - A forum can also be set up
  - The student may ask questions through a mic
- Roles and Responsibilities are not clear at this stage. However, several items are identified in this respect:

    - One takes the minutes, but the others should also take some notes. The Recorder makes a clean copy and distributes it via e-mail. Everybody should review these minutes and contribute if necessary.
    - The recorder of this meeting was Hakan Ozturk
    - The leader and meeting chair is Yavuz
- As time was short, nobody had something for the Proposal outline. It is agreed on that everybody collects related information for the Project Proposal Outline till Saturday
- As the next step:
    - On Saturday (09.10.2004) at 9.30 a new meeting will be held somewhere with a PC. The Proposal will be prepared.
    - Everybody is supposed to analyze existing systems related with the project (video streaming systems, e-learning systems, system development environments, etc.) and provide additional features from these systems as well as enhancements and possible ways of developing such a system
- The meeting was closed with a lunch…