Middle East Technical University

Department of Computer Engineering

System Requirements Specification and Analysis Report

by

1 Introduction

1.1 Project Background

As the technology evolves, the way that people get across had changed compared to as it was in ten years ago. Internet is intensively being used as a communication environment since 1995[1]. Today most of the people is using web to retrieve and send information since Internet is the largest and fastest way of this. Newsgroups and mailing lists have been the most popular approaches to this demand till now. After RSS 2.0 released in 2003, it has also become one of the most popular news feeding system around the Internet users. Besides there are still considerable amount of users using newsgroups and mailing lists. Since these methodologies use different conventions and nowadays news servers can not combine all of these conventions on their constitution, there's a serious need for an all-in-one news server that's synchronizing RSS, newsgroups and mail-groups.

1.2 Project Goals

The main purpose of this project is supplying a unified news server that has the ability of receiving and delivering threaded secure messages by using the common internet protocols and RSS. In other words our aim is to develop a news server system including the most popular communication protocols.

1.3 Project Scope

- Newsgroup, mail-list and RSS synchronization
- Secure data transmission
- Web-based Admin GUI allowing easy system configuration
- Built-in mailing list GUI for mail group trailing

1.4 Not in Project Scope

- Window-based Admin GUI
- Digital Signature

1.5 Project Objectives

- Build a server that can interact with a newsreader, mail client and RSS reader software
- Develop a core running as a web service and synchronizing the different built-in servers such as mail and news servers.
- Create a common user subscription mechanism for different access methods
- Build a RSS feed mechanism coherent with newsgroups and mail lists
- Develop a web-based Admin GUI for the configuration of whole system and manipulation of subscribed users

2 Market Research

When we decided to implement Unified Exchange News Server we had to make a study, especially on its basic structure and feasibility to be modeled.

The motivation for search was whether there exists a unified exchange news server in the market integrating NNTP newsgroups, web forums, blogs (RSS and ATOM feeds) and mailing lists. In other words whether the user can

- Post and read messages using any standard NNTP newsreader (Outlook Express and Thunderbird)
- Post and read messages through a web forum interface
- Post and read messages using a RSS/ATOM reader
- Post and read messages using any e-mail program with mailing list integration
- Post messages using any of the supported methods, and will instantly be able to read it with any other.

Güvercin is proposed to achieve this integration and additionally provide a web-based management panel with graphical user interface.

Since RSS and Atom is new technology, there is few news servers supporting RSS feed. Currently in the market there are traditional news servers which only provide Newsreader access and command-line server management. There are second group of servers integrating e-mail lists to news reader access.

2.1 Literature Review

2.1.1 DNews

DNews[2] is a widely-used traditional news server software. DNews is a news server which has a remote and local management panel with graphical user interface.

DNews can be extended by plug-ins.

- DNEWSWEB is a web to news gateway; it lets you merge your web pages and your news groups. By displaying USENET and local news groups on your web pages, users can read and post news directly using their favorite web browser.
- The DNews Admin Tool allows DNEWS to be managed locally or remotely from any windows 95/NT system. Both command-line management and management with GUI is provided .(locally and remotely)

PROS:

- Has a management GUI.
- Provides remote management

CONS:

- Is not free.
- RSS Feed is not provided.

2.1.2 INN

The InterNetNews[3] package (INN) is a complete Usenet system. It includes the InterNetNews daemon "innd", an NNTP server, and nnrpd, a newsreader server. INN separates hosts that feed you news from those that have users reading news. It is a freeware still being developed.

PROS:

- Free

CONS:

- Doesn't have a GUI.
- RSS Feed is not provided.

2.1.3 MUTANT PENGUIN (MPNews)

MPNews is a discussion forum solution for Windows. It helps groups of people communicate by whichever method they prefer by NNTP newsgroups, web forums, RSS and Atom feeds, and mailing lists[4].

PROS:

- It has a very-well designed Server Management Graphical User Interface which is very simple and easy to learn.
- Provides RSS and Atom feeds for your newsgroups, and lets you download other blogs into your newsgroups.
- Read and post messages to your newsgroups by e-mail.

CONS:

- It is not free.
- Remote management is not provided.

2.1.4 CONCLUSION

News Server	Newsreader Access	Web Access	RSS Feed	Atom Feed	Mailing lists	Management GUI	Web-based Management
Dnews	+	-	-	-	+	+	+
INN	+	-	-	-	-	+	-
MPNews	+	+	+	+	+	+	-
Güvercin	+	+	+	+	+	+	+

Table 1: Comparison of news server softwares

Table 1 summarizes the capabilities of the three news servers from the market and proposed capabilities of Güvercin. As it can be observed on the table MPNews is a powerful news server with four different access methods. However it is not free and it does not provide web-based management. Güvercin also will include these features.

2.2 Customer Survey

We have prepared our customer survey[5] by using an online survey provider[6] and news server users of METU Computer Engineering Department were chosen as the participants since we believe they have the ability of using news servers and they may show interest in our product. According to the results of the

survey most of the participants are interested in a news server that has the same capabilities with Güvercin.

2.3 Contacts

- **with Gökdeniz Karadağ**

Gökdeniz is an assistant in Computer Engineering Department at METU. As soon as we were assigned this project, we have contacted with him by email to tell us the general aspects of our business. The reason why we consulted with him was his tendency and background about networks. After Gökdeniz had accepted our meeting request, he arranged a seminar about the schema of our project. After giving brief information about RSS, he emphasized the following issues:

- The core system will act as a gateway between different types of clients and different types of servers.
- Storage of main components namely users and articles must be common, for they will be used both by built-in news and mail server.
- Security of the data transmitted between client and server can be supplied with using SSL
- Admin GUI of the server should be web-based. By this way admin configuration becomes platform-free.

- **with Halil Ağın**

Halil Ağın is a 2005 graduate of METU Computer Engineering Department. Since his interest area mainly based on networking and server systems, he was the first person we had contact with. After we have explained him the general structure of our project, we have asked him about the built-in server packages that we will use in our server. He advised us to use a web server, news server and mail server package in our system in order to lower the burden of core module. He has also stated that using a mailing list interface helps us about synchronizing newsgroups between news server and mail server.

- **with Yüksel Çalışkan**

Yüksel Çalışkan is the system administrator of a private company namely Enocta. After we have decided to use ready server packages for mail and news storage, we were eager to learn advantages and disadvantages of this approach and Yüksel Çalışkan was the right person to discuss this topic since he has been using such packages for more than 3 years. According to Mr. Çalışkan, the only difficult part of using such packages is the maintenance. He also stated that this disadvantage can be overcome with designing a comprehensive core system which can widely interact with these packages.

3 The Process

3.1 Process Analysis

When we faced with the problem of choosing process model of our project, we realized that linear incremental model is the most suitable one due to

the strict deadlines and preparing the reports once in general. In addition to that, we will have no real customer support and this will cause the lack of feedback of what we have done till that time. However our project's process model will also show characteristics of spiral model due to the weekly meetings with our supervisor Alev Mutlu, which will help us to overview our progress on a regular basis, and the fact that we will be writing two design reports.

3.2 Team Organization

3.2.1 Team Structure

As we stated in the project plan, our team do not have a permanent leader for a long time, one of us take the responsibility of organization according to tendency she/he has on the topic. Each member has exchangeable roles and we take our decisions based on our weekly meetings; communication among the team members is horizontal.

3.2.2 Ground Rules

- The project team will meet twice in a week; first meeting will be held on Friday afternoon and the second on Wednesday afternoon just after the meeting with our supervisor. At project proposal we stated different days but since we changed our meeting day with our supervisor our meeting days changed accordingly.

- Unless an acceptable reason is declared, missing the meeting will not be tolerated.

- Late attendance to meetings will not be tolerated.
- Each member should attend to the meetings well-prepared.
- On Friday meetings, weekly project task of each individual will be decided; furthermore if a task requires group work, the structure of group(s) will be arranged.
- On Wednesday meetings each member will inform the others about his/her weekly assignment. The divided tasks of that week will also be compiled in this meeting.
- At the end of each Wednesday meeting the agenda of the next meeting will be decided.
- In each meeting, minutes will be taken by the recorder of the week.
- All members are responsible for checking the e-mail group daily.

3.3 Major Constraints

3.3.1 Hardware Constraints

As we stated in hardware requirements section, our product needs may not work efficiently on Desktop Systems. Since the project team consists of home users, we may not have the chance to test development of our project frequently.

3.3.2 Software Constraints

After our investigations from literature review, we found out that there is not a compact news server development environment in the market. Moreover some tools that we can use for developing several parts of the project are not open source. So we may face with the problem of finding a common working space while we are implementing Güvercin.

3.3.3 Experience Constraints

We are doing this project in a group of five and none of us has an experience of such a large project. Moreover, none of us has knowledge of network programming. These directly affect the analysis and design of the project.

3.3.4 Schedule Constraints

Under the pressure of the near deadlined projects and heavy workload of our department, unfortunately, we will not always have flexible time to spend on our project. In order to complete this project at the end of this year, we must obey the strict deadlines of the milestones.

4 Requirement Analysis

4.1 System Requirements

4.1.1 Hardware Requirements

Although most of the news servers best perform on server systems, our product Güvercin performs on the following system requirements:

Pentium 4 main processor, 512 MB of RAM, modem or network card.

4.1.2 Software Requirements

Güvercin can perform on both Linux and Windows systems and no additional software is required.

4.1.3 Development Requirements

Güvercin will need the following technical requirements in order to be developed:

- **Perl Express 2.5:** Perl[7] Express is a free perl development environment. We will use it for implementing the core part which will act as a gateway between built-in servers and clients.
- **Macromedia Dreamweaver MX 2004**: As we will implement admin GUI by using PHP, Macromedia Dreamweaver is the best choice for designing a comprehensive admin tool.

- **Apache Web Server**: Apache[8] will be used as our HTTP Server because it is compatible with the other tools we will use.
- **InterNetNews Daemon:** Our unified news server needs a built-in NNTP server in order to serve NNTP clients. So we decided to use the most common one namely INND[9].

4.1.4 Product Requirements

After we complete our project, the following requirements must be meet by our product:

- Platform independence
- Secure data transmission
- Developer-friendly
- User friendly admin configuration

4.2 Functional Requirements

The interaction between clients and built-in servers is the first major requirement of our project. For such a news server there are four main client types:

- SMTP Client
- NNTP Client
- RSS Client
- HTTP Client

RSS Client directly refers to an URL on a HTTP server. HTTP Client also uses HTTP Server but there exists a converter between HTTP server and Core in order to represent the HTTP Client as an NNTP client. Therefore first main requirement is building a core acting as a gateway which can serve both NNTP and SMTP clients.

For SMTP and NNTP are two different communication mechanisms, they use different conventions for keeping the message while it is being transferred. If a server wants to handle NNTP and SMTP clients, the main requirement is the synchronization of these protocols at three different levels.

- **Read-Level Synchronization:**

  If an NNTP client sends a read request to the server, since it can not be both SMTP and NNTP client at the same time, there's no need for NNTP to SMTP conversion. Same approach can also be used for a SMTP Client. Lastly for RSS Clients the .xml files must be kept in a file storage managed by a HTTP server.

- **Write-Level Synchronization**

  If an NNTP or SMTP client wants to post a message to the server, first it must be kept in the cache of our core. First attempt must be delivering the message to the related server. After that one of NNTP2SMTP or SMTP2NNTP conversion must be made for delegating the same message to the other server. Finally if the newsgroup which the message belongs to gives an RSS feed same message must be converted into an xml file and sent to the file storage handled by the HTTP server. If any one of these

attempts fails, the taken actions must be revoked and a failure notification must be returned to the client.

- **Update-Level Synchronization**

  Since only an NNTP client can have a update request, there is no need to encounter this situation for other type of users. When such a request comes, the first thing that the core must perform must be delegating this request to the news server. The next step is converting the updated message to SMTP format. Since a mail can not be updated, updated message must be sent as mail to the subscribed users of the corresponding newsgroup. After all, if newsgroup that contains the message is also used as RSS feed, related xml file must also be updated. If any of these attempts fails, the taken actions must be revoked and an error must be returned to the client.

The second requirement for our project is user delegation. Our system will use SSL for security and it requires a session established between the user and our built-in news or mail server for secure interaction. If we use a gateway to transmit the message coming from one of SMTP or NNTP client, such a session can not be established between sides just because there's no direct connection. So a need occurs for a core system which can establish this session with the suitable server on behalf of the client. So each command and data coming from these types of clients requires a delegation by the core for security reasons.

The way that the admin communicates with the system is also a major topic for our project. Following requirements must be encountered:

- Admin can create, edit and delete newsgroups.
- Admin can create, edit and delete mail groups.
- Admin can add and delete users.
- Admin can configure core, news server, web server and mail server.

Although we have mentioned about the major requirements for our system to function user requests, there are also a few more points that our system must encounter. If we compile them together, users of GÜVERCİN have the capability of

- Reading and posting articles
- Configure their mailing list settings by using we interface
- Getting the RSS feed for some newsgroups

5 Use Case Analysis

5.1 Use Case Diagrams

5.1.1 Use Case for Admin

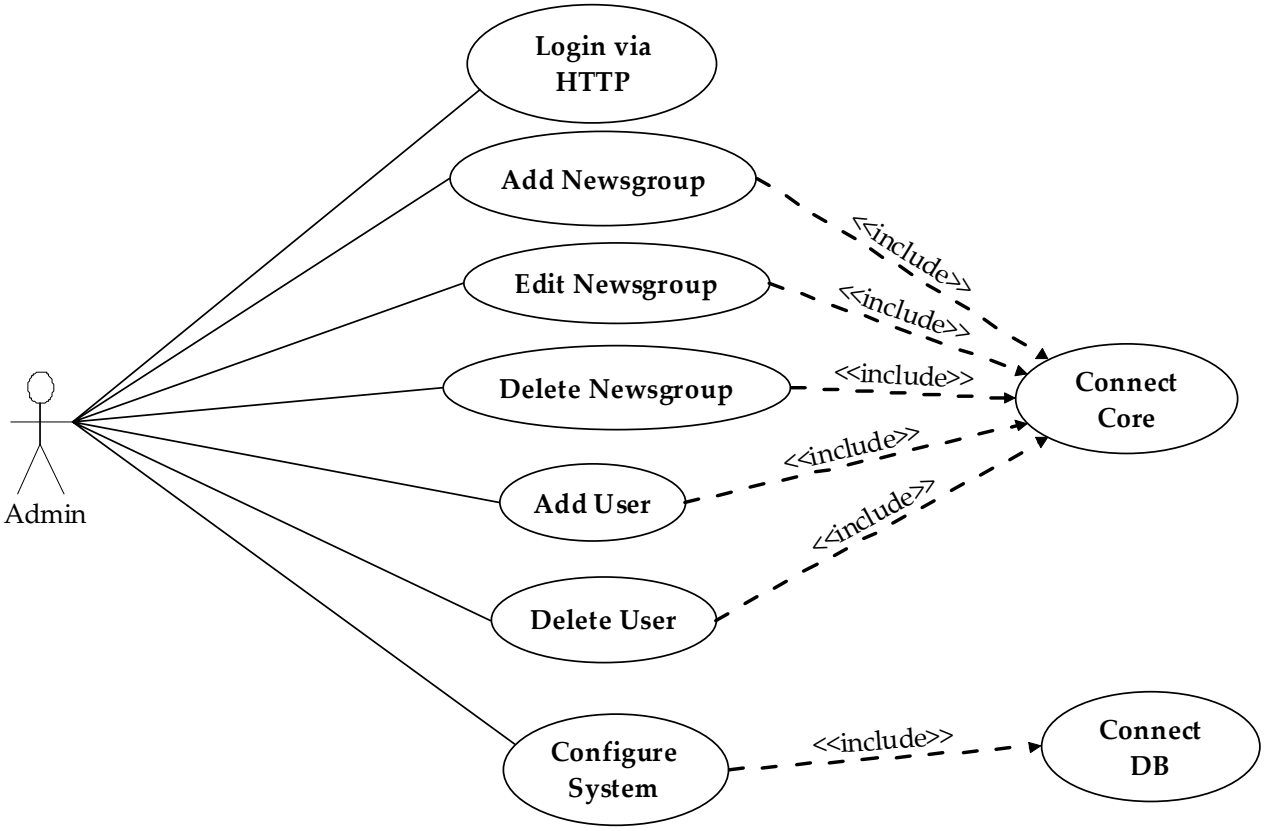

Figure 1: Use Case for Admin

5.1.2 Use Case for Subscribed User

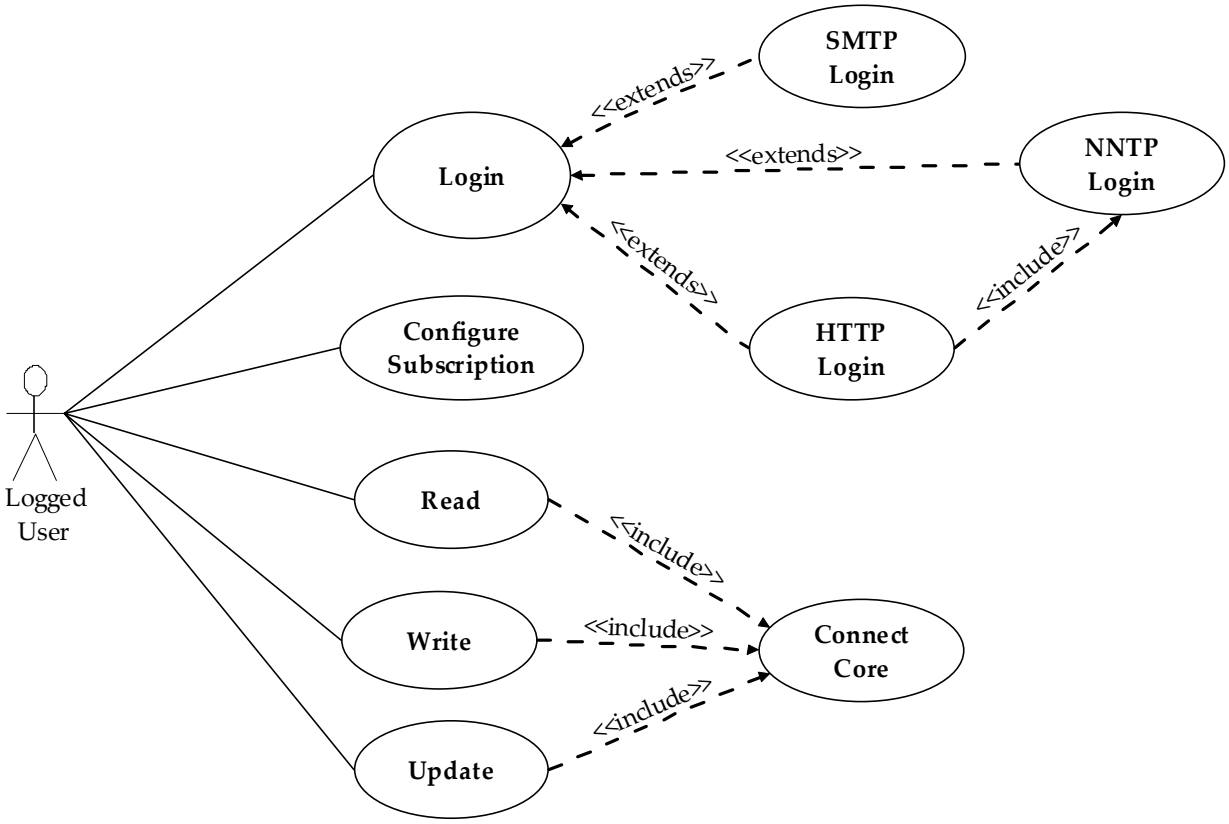

Figure 2: Use Case for Logged User

5.1.3 Use Case for RSS User

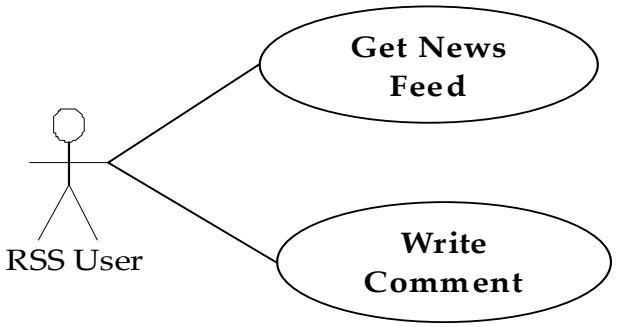

Figure 3: Use Case for RSS Client

5.1.4 Use Case for User

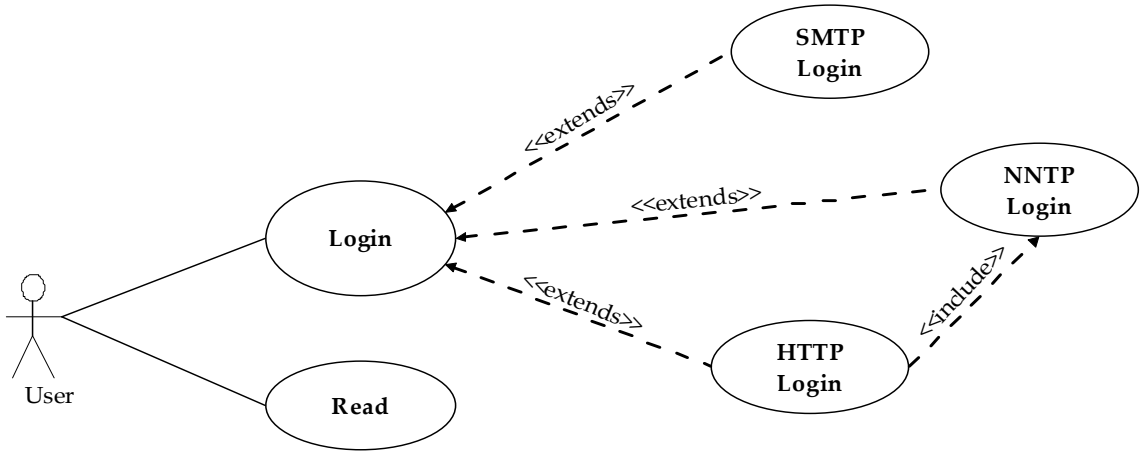

Figure 4: Use Case for Naïve User

5.1.5 Use Case for Core

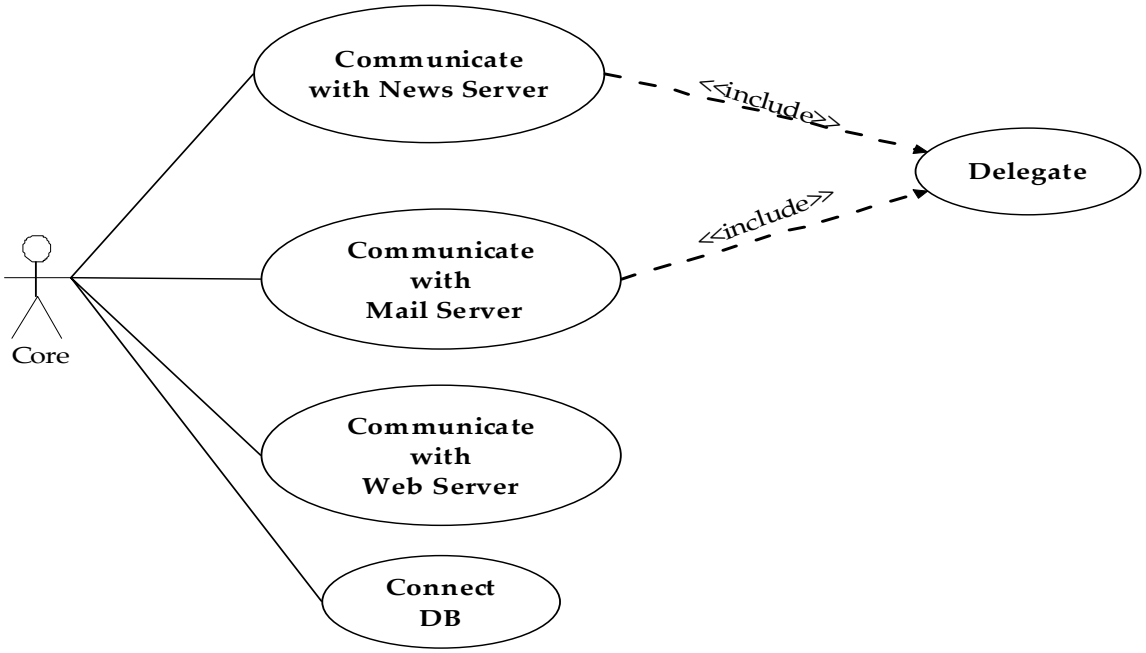

Figure 5: Use Case for Core

5.2 Use Case Scenarios

5.2.1 Use Case Scenario for Admin

1. Admin shall login to the system by using form lying in the web GUI.

2. After a successful login, he will choose one of the '*Manage Newsgroups'*, *'Manage Users'* or *'Configure System'* option.

3. In order to *'Manage Newsgroups'* or *'Manage Users'*, he/she has to '*Connect Core*'.

4. By using the core connection, he/she has the right of doing operations on users and newsgroups such as editing, deleting and adding a new one.

5. For configuration of the core, admin shall connect to the database by using the web server.

5.2.2 Use Case Scenario for Subscribed User

1. User shall *'login'* to the system by using the form lying in the web user interface of mailing list system.

2. After a successful login, the user shall *'read'* and *'write'* articles and *'update'* his/her own articles (for NNTP Users) of the related newsgroup and shall *'configure subscription'*. (for mailing list users).

5.2.3 Use Case Scenario for RSS User

An RSS User does not have to be a subscribed user in order to '*read RSS'* and '*write comment'* to them. So he/she just simply sends a request and gets reply.

5.2.4 Use Case Scenario for User

1. A naïve user shall '*login*' to the system by using one of '*NNTP Login', 'SMTP Login'* or '*HTTP Login*'

2. A naïve user shall *'read'* newsgroups without login operation. (Visitor case)

5.2.5 Use Case Scenario for Core

1. Core shall *'Communicate with News Server'* by *'delegating'* the user's request.

2. Core shall *'Communicate with Web Server'* for writing RSS, serving HTTP Client's requests and admin configuration.

3. Core shall *'Connect DB'* for reading user information (username, password) and admin configuration.

6 Modeling

6.1 Data Modeling

6.1.1 Entity Relationship Diagrams

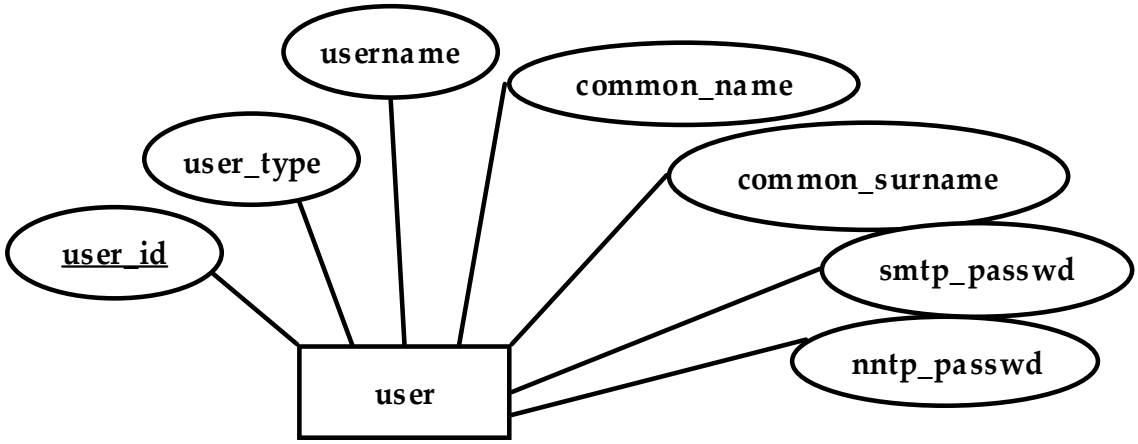

Figure 6: User Entity

The core will be establishing a session with NNTP Server and Mail Server on behalf of the user which will be delegated. Thus, we have to have a user entity in our user and core configuration data store.

Each user has a unique *user_id* which will also be the key of our user_id. This will be assigned to each user by the system. In addition to that each user will have a unique *username* that will be used throughout the whole system as the screen name.

Mainly, there two types of users; subscribed users and administrators. The type of each user will be kept in the *user_type* attribute of the user. *common_name*

and *common_surname* will give us the full name of the user. *smtp_passwd* and *nntp_passwd* will allow the core to establish the secure connection.

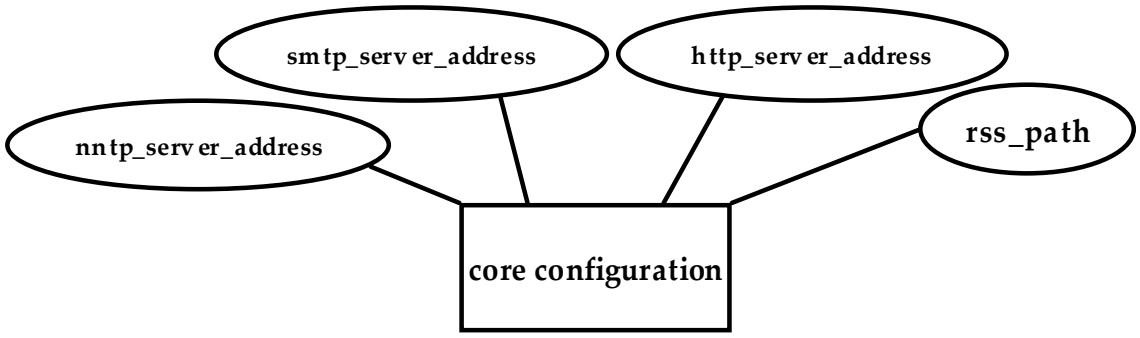

Figure 7: Core Configuration Entity

The core configuration entity has the attributes, *nntp_server_address, smtp_server_address*, *http_server_address* and *rss_path* for our system will have the capabilities of each of them and to have those capabilities we have to configure the core with these addresses. Each of these attributes will be stored as VARCHAR data type. Admin will have the capability of configuring whole system by using entities of this table.

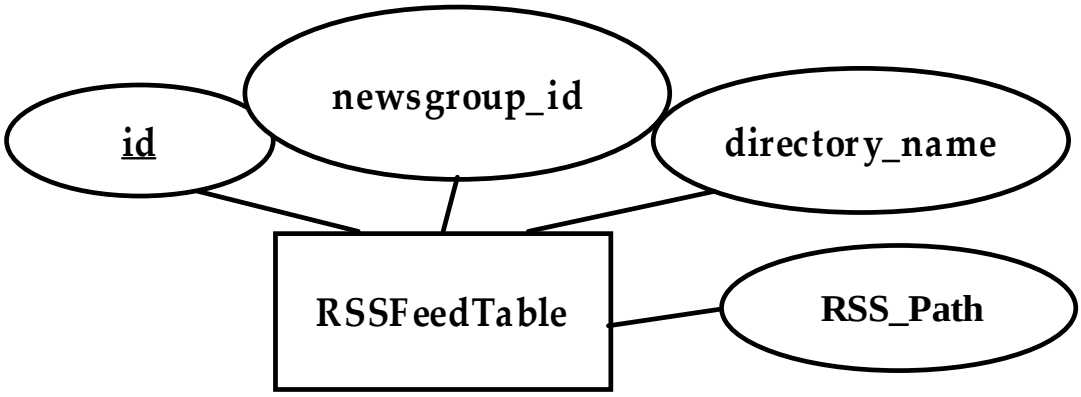

Figure 8: RSS Feed Table Entity

Each RSS feed in the RSSFeedTable will have a key attribute id that the system will give. The RSS feeds which will be given by the newsgroups will be arranged by the administrator user. Administrator can also change the directory of the xml files used as RSS feeds by using *directory_name* entity in the table. Also URL of the each news feed is stored by using *RSS_Path* entity.

6.2 Functional Modeling

6.2.1 Data Flow Diagrams

Our level 0 data flow diagram seen in the figure 1.0 reflects the general behavior of GÜVERCİN. According to this diagram user can interact with our unified news server in several ways.

6.2.1.1 Level 0

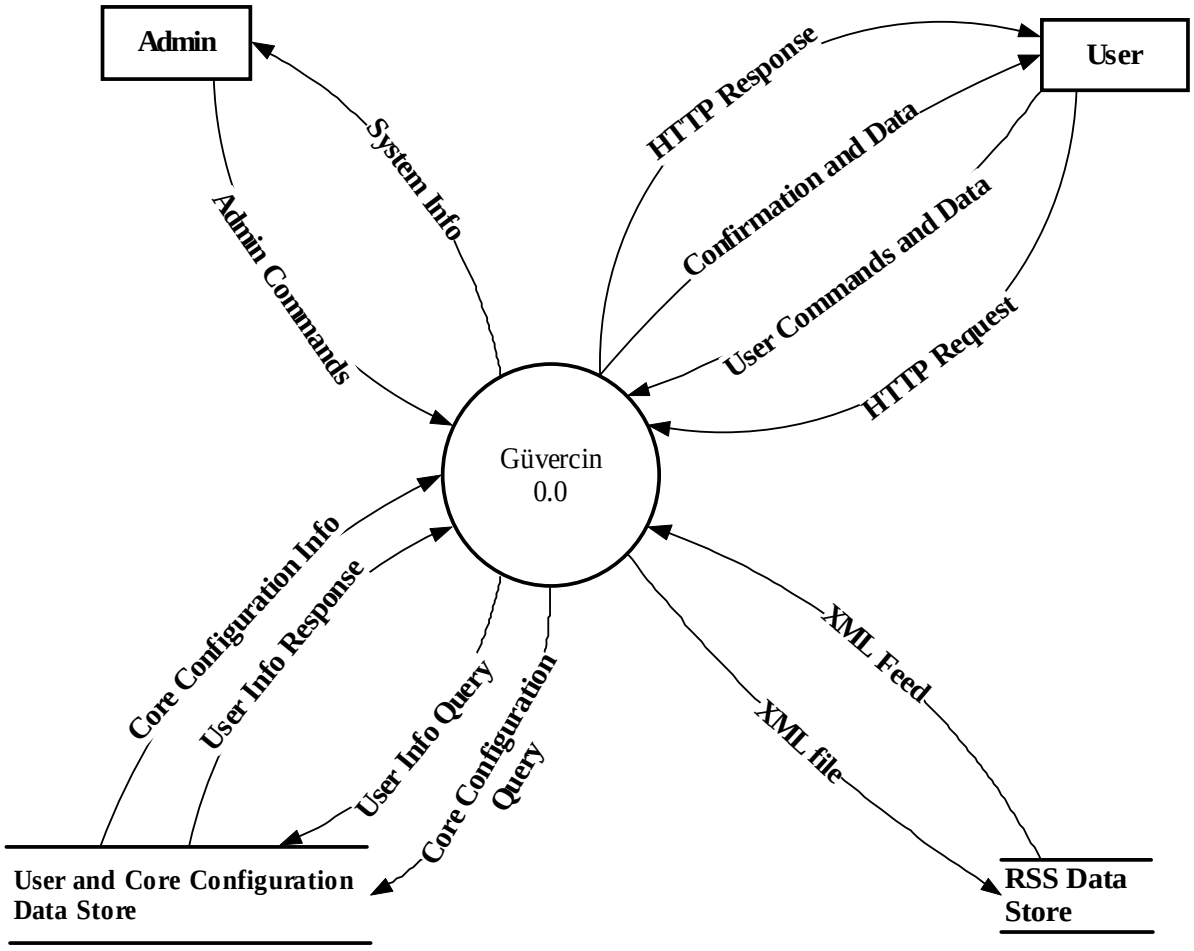

Figure 9: Level 0 DFD

6.2.1.2 Level 1

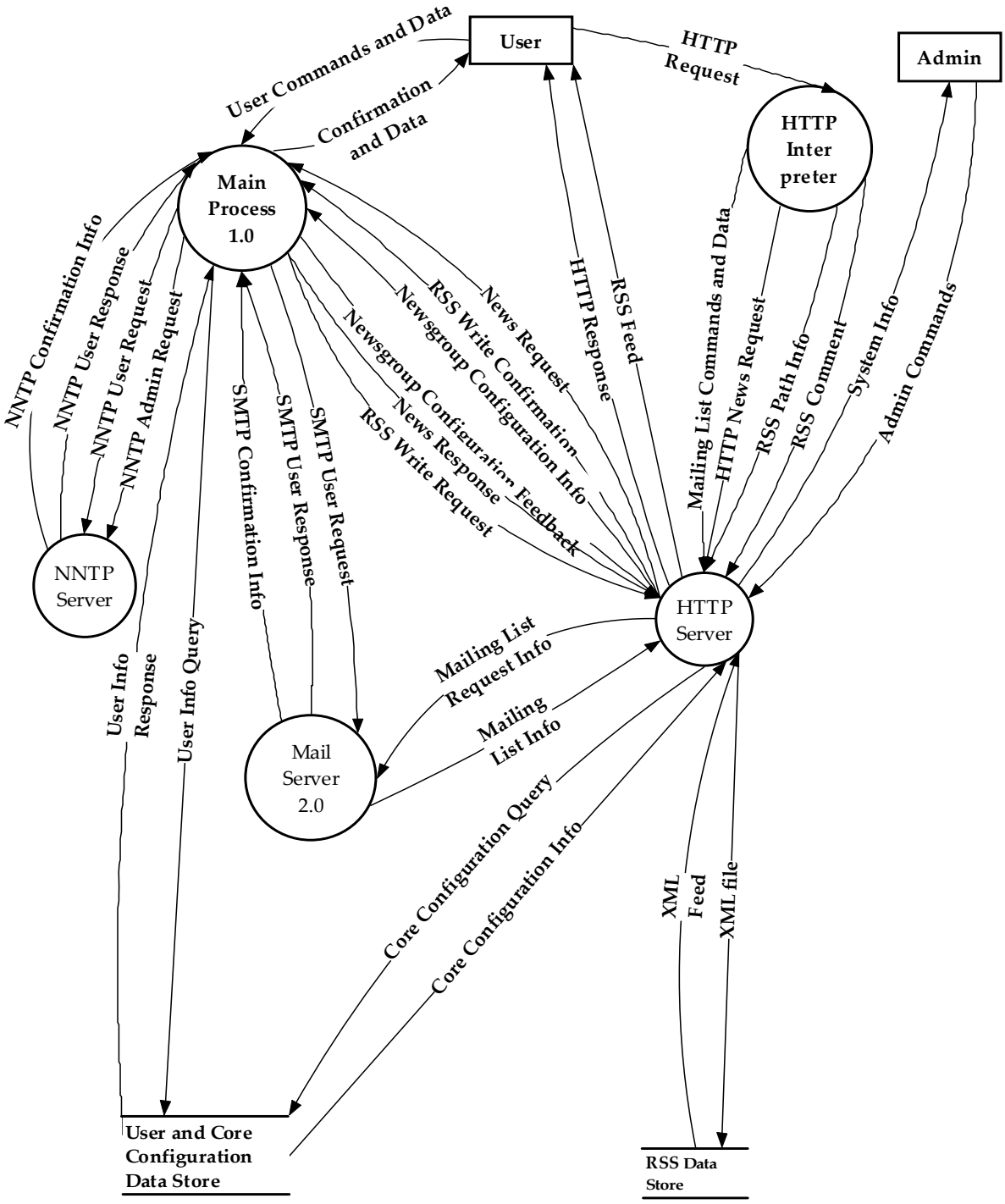

Figure 9: Level 1 DFD

6.2.1.3 Level 2

6.2.1.3.1 Level 2 DFD for Main Process 1.0

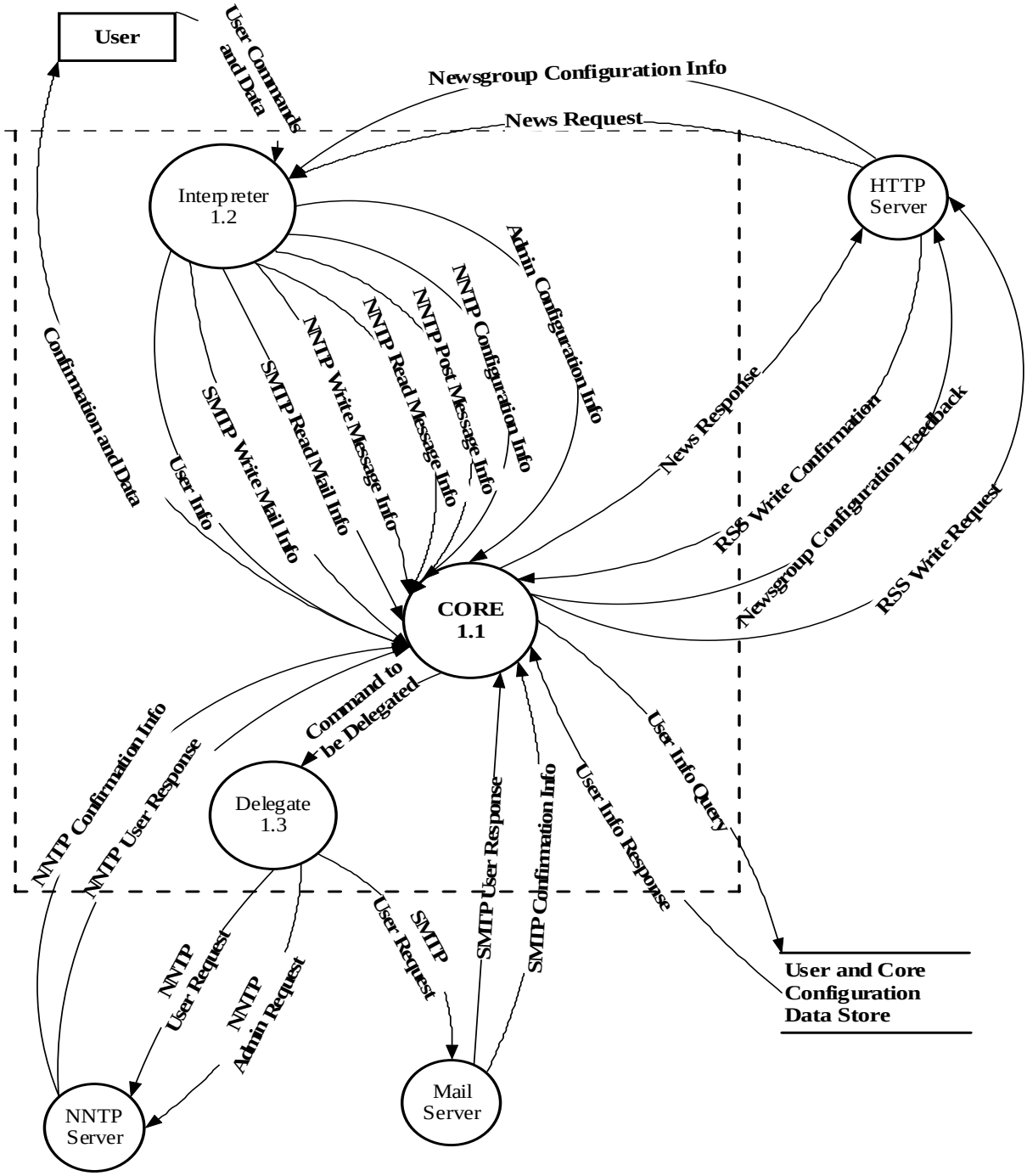

Figure 10: Level 2 DFD for Main Process 1.0

6.2.1.3.2 Level 2 DFD for Mail Server 2.0

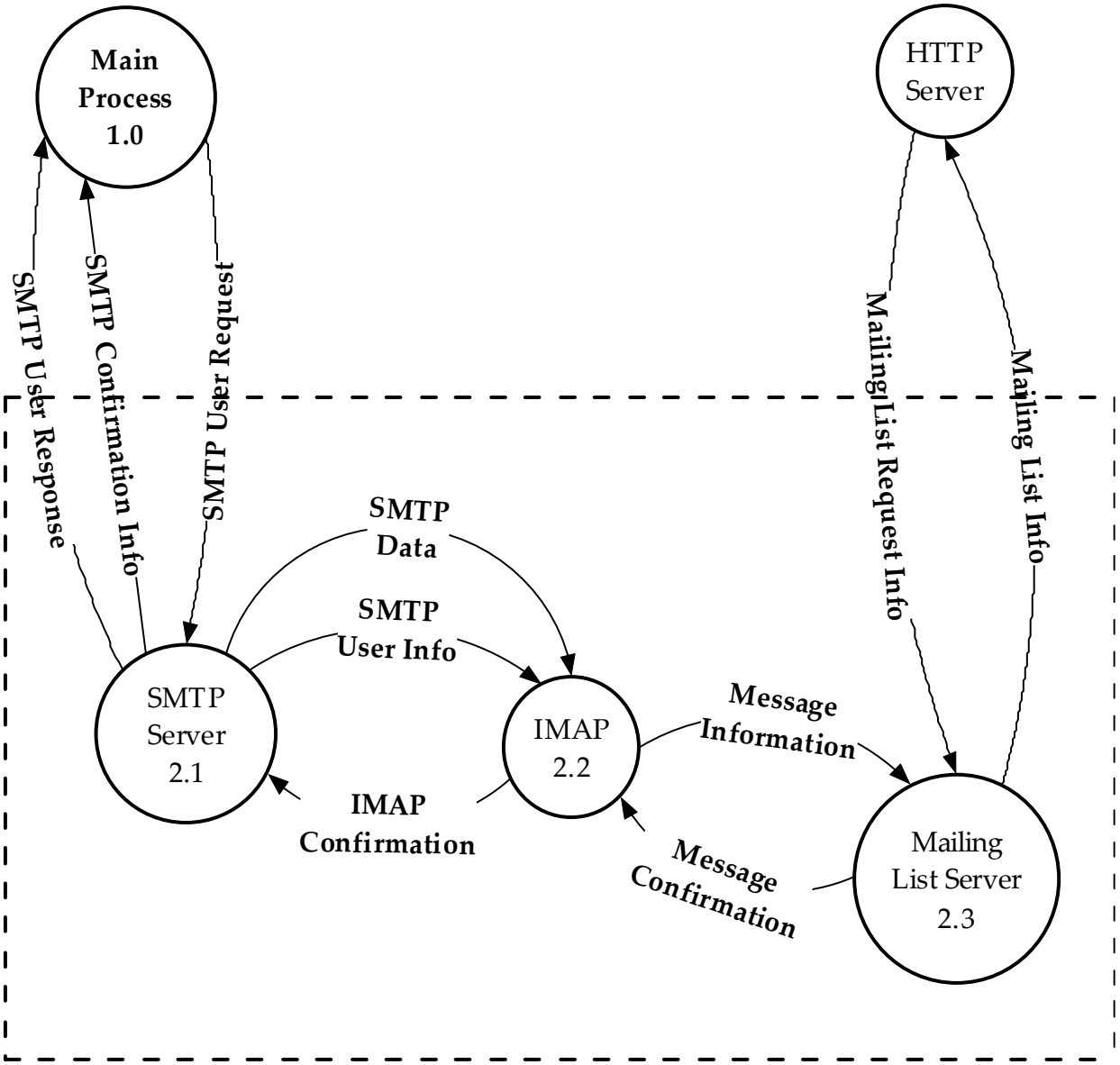

Figure 11: Level 2 DFD for Mail Server 2.0

7 Project Schedule

7.1 Project Milestones

Below, we stated the work packages of our project:

WP1. Project Management

- Project Proposal

- System Requirements Specification and Analysis Report

- Initial Design Report

- Final Design Report

WP2. Core

- User Authentication and Security Implementation

- NNTP2SMTP Module Prototype Implementation

- SMTP2NNTP Module Prototype Implementation

- Prototype Integration and Testing

- Prototype Release

- NNTP2SMTP Module Complete Implementation

- SMTP2NNTP Module Complete Implementation

- RSS Module Implementation

- Core System Integration

- Alpha Testing

- Beta Testing and Debugging

- Core System Release

WP3. Admin Panel

- GUI Design and Implementation

- Panel Implementation

- Alpha Testing

- Beta Testing and Debugging

WP4. Finalizing Activities

7.2 Gantt Chart

Our Gantt Chart can be seen in the Appendix.

8 Risk Management

8.1 Risk Table

RISK	**PROBABILITY**	**IMPACT**
Lack of Knowledge and Experience on Subject	**%40**	**1**
Lack of Experience on Conducting Project Management Process	**%25**	**2**
Problems Due to the Team Members	**%5**	**4**
Failure in Performing Due Responsibilities and Unavailability of a Team Member	**%25**	**2**
Time Problems Due To Unrealistic Scheduling	**%10**	**3**
Lack of Requirement Analysis	**%30**	**2**

Table 2: Risk Management Table

8.2 RMMM Plan

8.2.1 Scope

Risk analysis and management are a series of steps that help a software team to understand and manage uncertainty. A risk is a potential problem it might happen, it might not. But, regardless of the outcome, it is a really good idea to identify it, asses its probability of occurrence, estimate its impact, and establish a contingency plan should the problem actually occur. Thus; for the safety of our project we decided to perform some risk management activities through the life cycle of the project. RMMM plans and risk table which will be shown in the preceding sections are not the final documents and will be updated in next versions in case of existence of new identified risks.

8.2.2 Risk Mitigation, Monitoring and Management (RMMM) Plan

- **Lack of Knowledge and Experience on Subject**

  Since we have not taken a network course in our department and none of the team members have worked on network and security areas before; suffering from knowledge and experience is a high possibility. In order to handle this problem we decided to spend considerable time on research at the very beginning of the project. Also since we do not have the opportunity of hiring an experienced member on these topics, we follow the way of consulting on a person, whose interesting area is network related things, about critical topics of our project.

- **Lack of Experience on Conducting Project Management Process**

Having mostly done less complicated and individual homework till this semester in our department group-work projects are so strange for all group members. Besides the advantages of working in a 5-member team an taking the decisions together we try to predict the possible disadvantages and stated that coordination and communication are the two critical problems that we have to overcome in order to carry on this project. So first we set 2 meetings every week and state strict rules for attendance. Also we form a mail-group and decided to discuss the project based topics at that group.

- **Problems Due to the Team Members**

Although not expected, withdrawal of a team member from the project and possible disagreements between group members can affect our usual work flow. In order to prevent such situations all group members agree on taking professional attitude to the job and try to keep separate this project from his/her social relationships between other group members. Also we want all team members to post his/her works to our mail-group regularly in order to have an archive in case of a withdrawal. Handling of possible disagreements and solving the problems are left to the regular weekly meetings since face-to-face discussion is much more effective to solve the problems rather than discussing it in the mail-group.

- **Failure in Performing Due Responsibilities and Unavailability of a Team Member**

Since all group members are senior students and taking some elective courses besides this project there will be times of collision of deadlines

happen. This is another case which we try to overcome with the help of our regular meetings. At the end of each meeting while assigning the tasks to the team members; each member is responsible for notifying his/her situation to the group if there can be a problem with his/her workload that week. If possible workload of that group member but he/she will be warned in case of repetition. If there is no chance for regulating the workload, extra effort asked from that group member for a week.

- **Time Problems Due to Unrealistic Scheduling**

  All the possible cases considered and an acceptable scheduling expected to be arranged at the very beginning of the project. But because of some reasons such as lack of experience etc. we may overestimate our capacity and set up an unreasonable schedule. For this reason we choose Yusuf Saran responsible for following the works that we have done weekly and adjust our situation according to the schedule. Also if the schedule is so unrealistic to follow, then we arrange an urgent meeting and decide on a new and more reasonable schedule and try to follow it now then.

- **Lack of Requirement Analysis**

  Misunderstanding of customer requirements can be a great problem in this kind of a project. To avoid this risk, we arrange weekly meetings with our supervisor and take notes of these meetings carefully. Also we try to meet people from as many companies as possible. Before going to those meetings we arrange a group meeting and decide the questions that we can ask. At meetings we want them to be as clear as possible and ask the

questions that we have already stated. Also we ask them feedbacks at regular intervals.

9 Appendix

[1] inet-tr.org.tr/inetconf8/bildiri/105.doc

[2] http://www.netwinsite.com/dnews.htm

[3] www.isc.org

[4] http://www.mutantpenguin.co.uk/mpnews/

[5] www.freeonlinesurveys.com

[6] http://freeonlinesurveys.com/rendersurvey.asp?sid=yl8p4q3h6v0y8i0237140

[7] http://www.perl-express.com

[8] www.apache.org

[9] http://www.eyrie.org/~eagle/software/inn/docs/innd.html

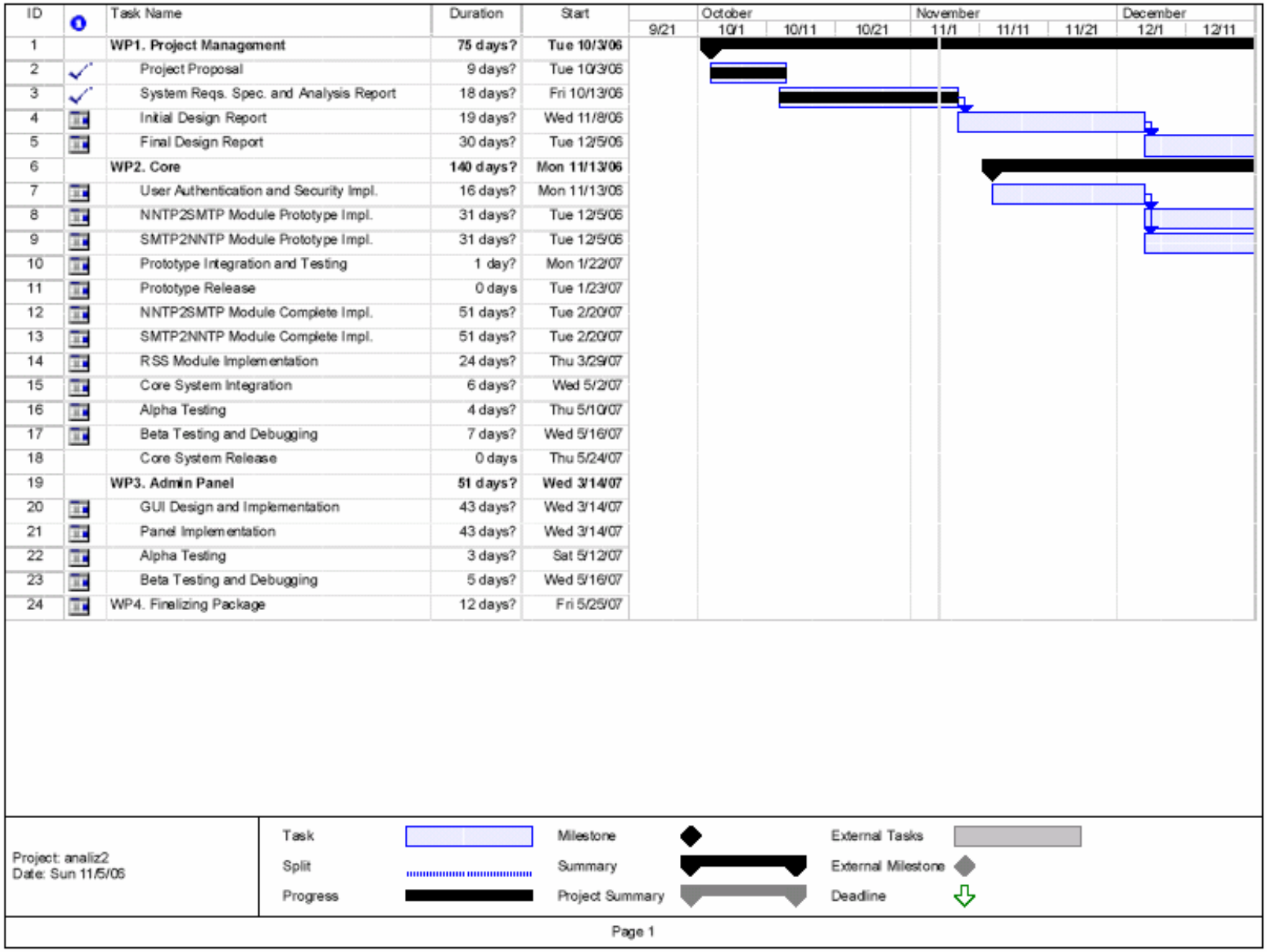

Figure 12: First Part of the Gantt Chart

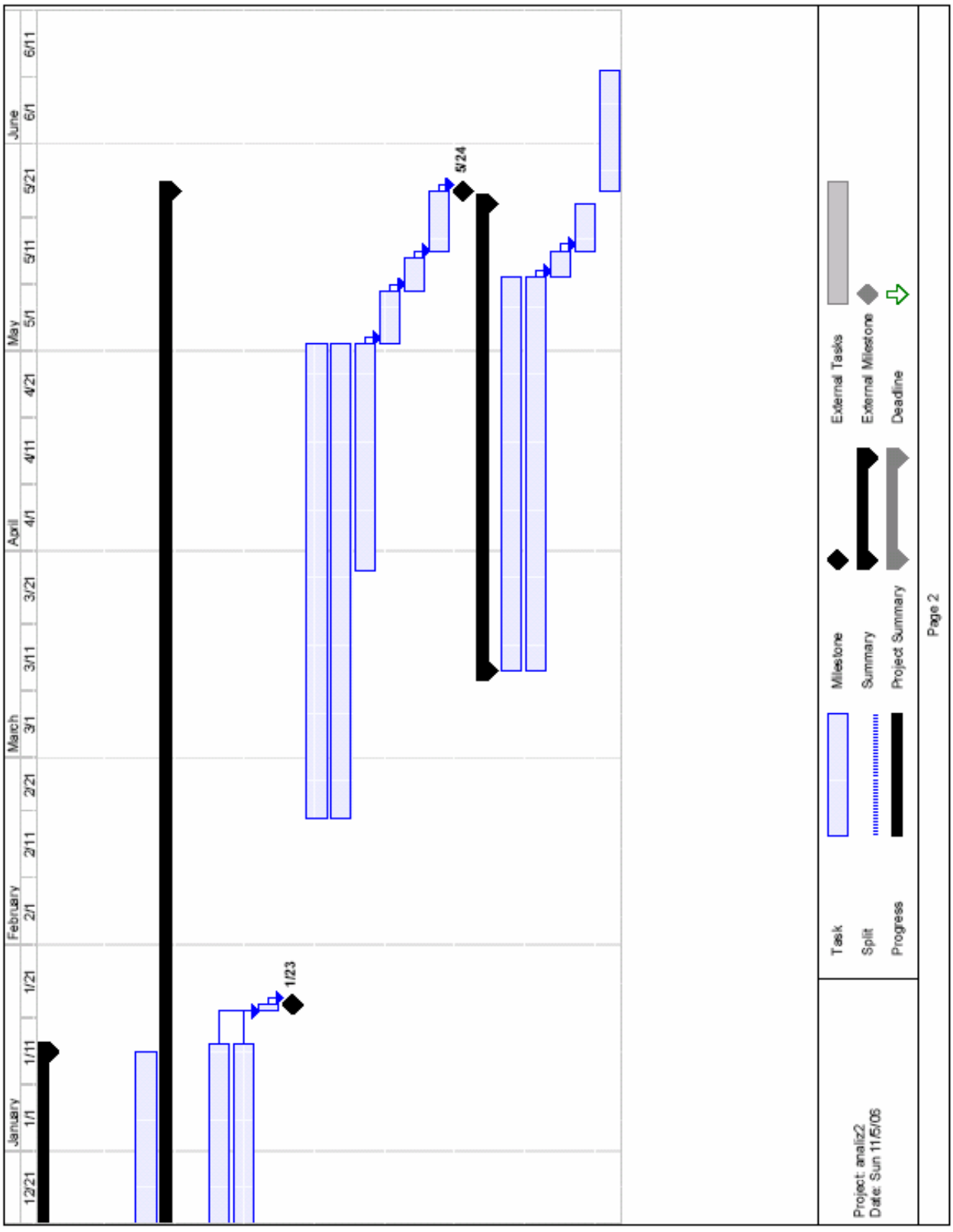

Figure 13: Second Part of the Gantt Chart