MIDDLE EAST
TECHNICAL UNIVERSITY

DEPARTMENT OF COMPUTER ENGINEERING

NewStreamLine

SYSTEM REQUIREMENT SPECIFICATION
AND
ANALYSIS REPORT

Ainura MADYLOVA	**e1408657**
Asiye KIYAK	**e1347673**
Fatma ÖZYILDIRIM	**e1347830**
Esin YÖNDEM	**e1348168**
Hüsna IŞIK	**e1347566**

TABLE OF CONTENTS

1. Introduction

1.1 Purpose

Purpose of this project is to write the universal server for news exchanging which will give to user an ability to rich the news via different types of Internet protocol such as HTTP, SMTP, NNTP using the secure data transfer layers.

1.2 Scope and Project Description

We live in a dynamic world which continuously integrates all kind of innovations with our daily life. What remains the same is the wish of mankind to communicate with each other. With the invitation of Internet these facility become very popular and shifted it to higher level. NewStreamLine is the environment that aimed to provide its users with all features of that level. It merges Mail Service, News Service and Forum-like Web-Service for fast and safe information exchange.

Main part of the project is implementing the middleware named Core in the *Figure 1* which represents the architecture of the system.

No server has a direct access to the database. All communication is done through the Core. Core is divided into 3 modules. First module always listen to the ports connected to the Mail Server, News Server by continues loop, if any request comes from the servers, Core does the operation (for example retrieving data from the database) and sends response to the servers using appropriate protocols. Also RSS format is generated depending on user's request. All other communication between the system and external services is done through this module. Second Module is Web Service, which manages all web application (for example a forum site) of the system. Again, there is no direct access to the database that's why web service is the part of the Core. The last module is the SQL/XML Engine module, which interacts with database. This module listens for the triggers that comes from database and communicate with remaining modules continuously. So there is a main loop in the Core, which waits for modules to communicate with each other and small loops which listen to the external ports and send the information through them if needed. The aim of the project is to make all services work together and Core does this.

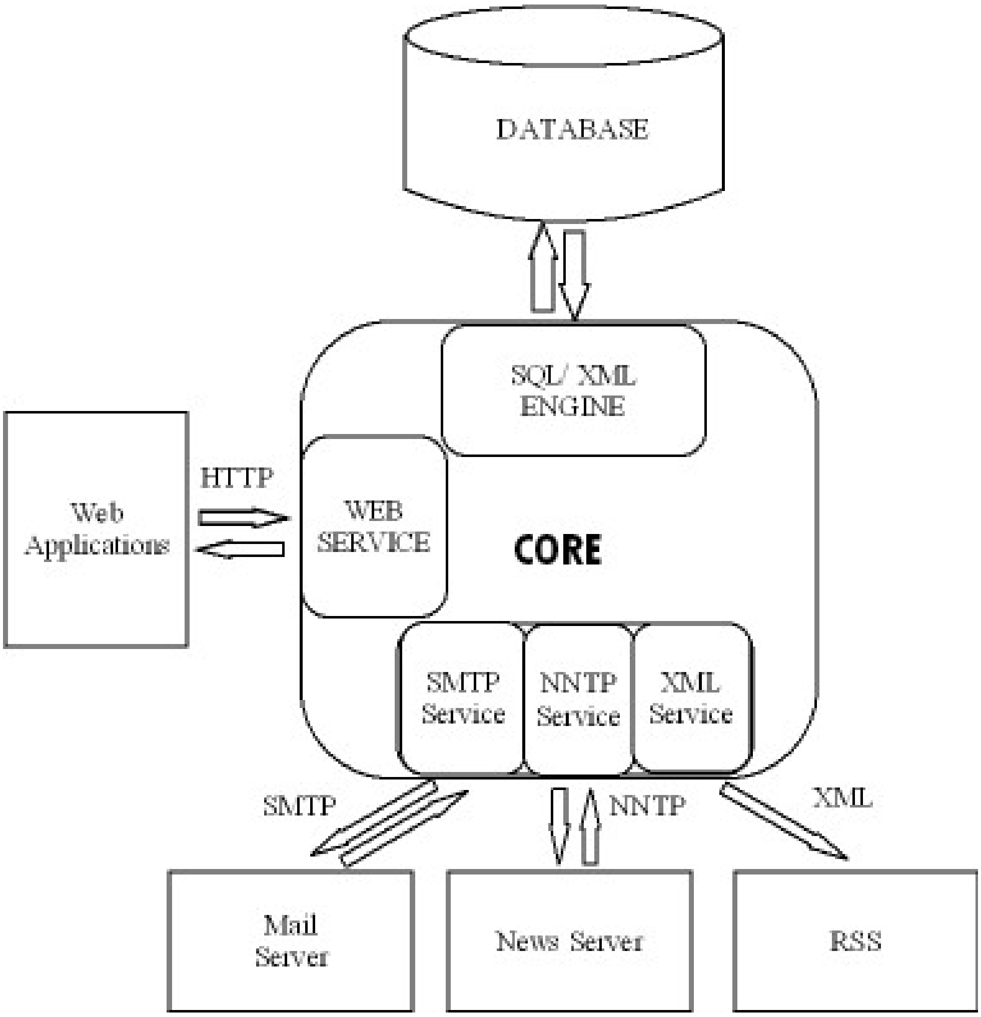


Figure 1

The second part of the project is constructing web application that will represent the system. From this point on system will also be called as NewStreamLine. Although the concept and construction is easier in this part, the quality of the web application plays a big role, because the forum page of the NewStreamLine will be a “face” of it and all interaction between user and the system will be done through it.

Users will have ability to:

- Sign Up (create a new account) to NewStreamLine
- Log In using Username and Password

- Create new topics/ Create new groups/ Subscribe/ Unsubscribe in a News Group
- Post/Replay to/Discuss messages
- Choose articles to be send to E-mail/ Manage the sending process
- Change Account Information
- View the statistics of topic reading
- Choose RSS file format to view news topics
- Search topic in the system

Secure Sockets Layer (SSL) and Transport Layer Security (TLS) will be used for secure of data transfers.

2. Process

2.1 Team Structure

Topics related to the project are new for all of us, and because of this we decided to choose Democratic Decentralized (DD) team structure for our team. Each member will be working in particular area and be fully responsible of it. In the weekly meetings every member will explain their researches and suggests solutions for the problems related to their topics.

2.2 Process Model

Also we have predefined deadlines for delivery of every step of our project we decided to choose iterative model for it. Most of the parts in the project are very new to all of us that's why understanding everything in the beginning is nearly impossible and there of course would be some changes in the steps done before. Clearly, the research and learning stage will continue through out whole project. By choosing iterative model we will always review previous reports and correct them if needed. Feedback reached in the end of the process will help us doing this.

3. System Requirements

Requirements given in this section are flexible since the number of system users and system load differentiate depending on the use area of our product.

3.1 Hardware Requirements

NewStreamLine Server's working principle is like a web server; it will never be shut down and give nonstop service to its clients. Depending on the system usage, there must be an

Internet connection fast enough to accept the daily load; and two HDD's to be used for taking backups of the system.

The expected number of system users is approximately 100 clients. For each user 10 MB and for the administrators 40 MB free disk space will be preserved in the server side. For this amount of space and number of users, our own computers are satisfying our needs. However, if the system is going to be used intensively by thousands of users, the carriage of the load would be much more difficult. For example, disks must be able to support adding very high number of new files per minute, the directory accesses, file creation, write and file closes to keep up with the feed, etc.

3.2 Software Requirements

We intended to use NNTP, SMTP, HTTP, RSS and a web server for our project. We researched among many web servers and decided to use the most popular and advantageous Apache Web Server. Moreover, we decided to use MySQL as a relational database management system. We give importance to our product's platform independence. Since Java technologies are platform independent and compatible with both Linux and Windows, we are going to use Java programming language.

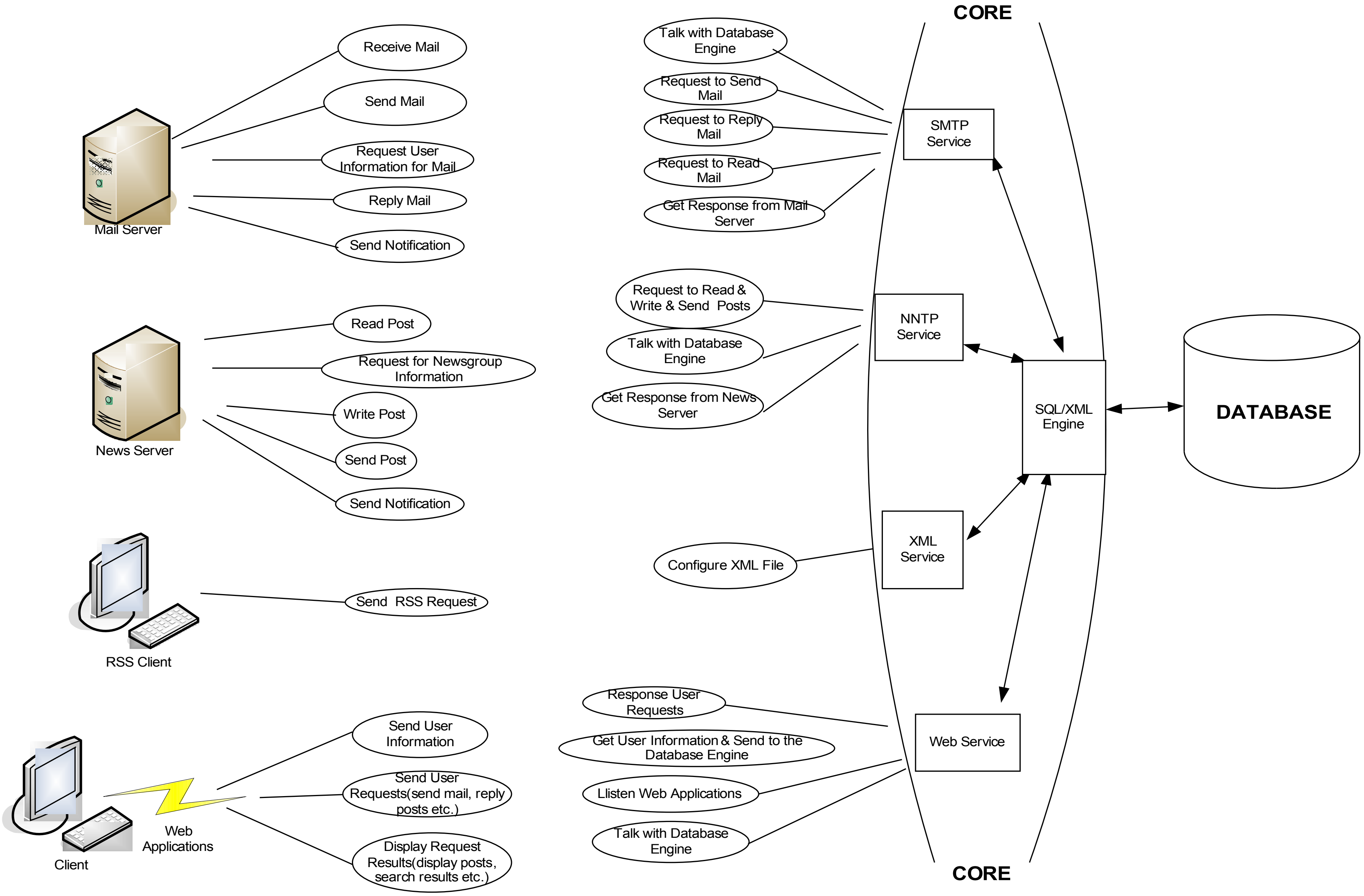
4. User Requirements

4.1 Use Case Diagrams

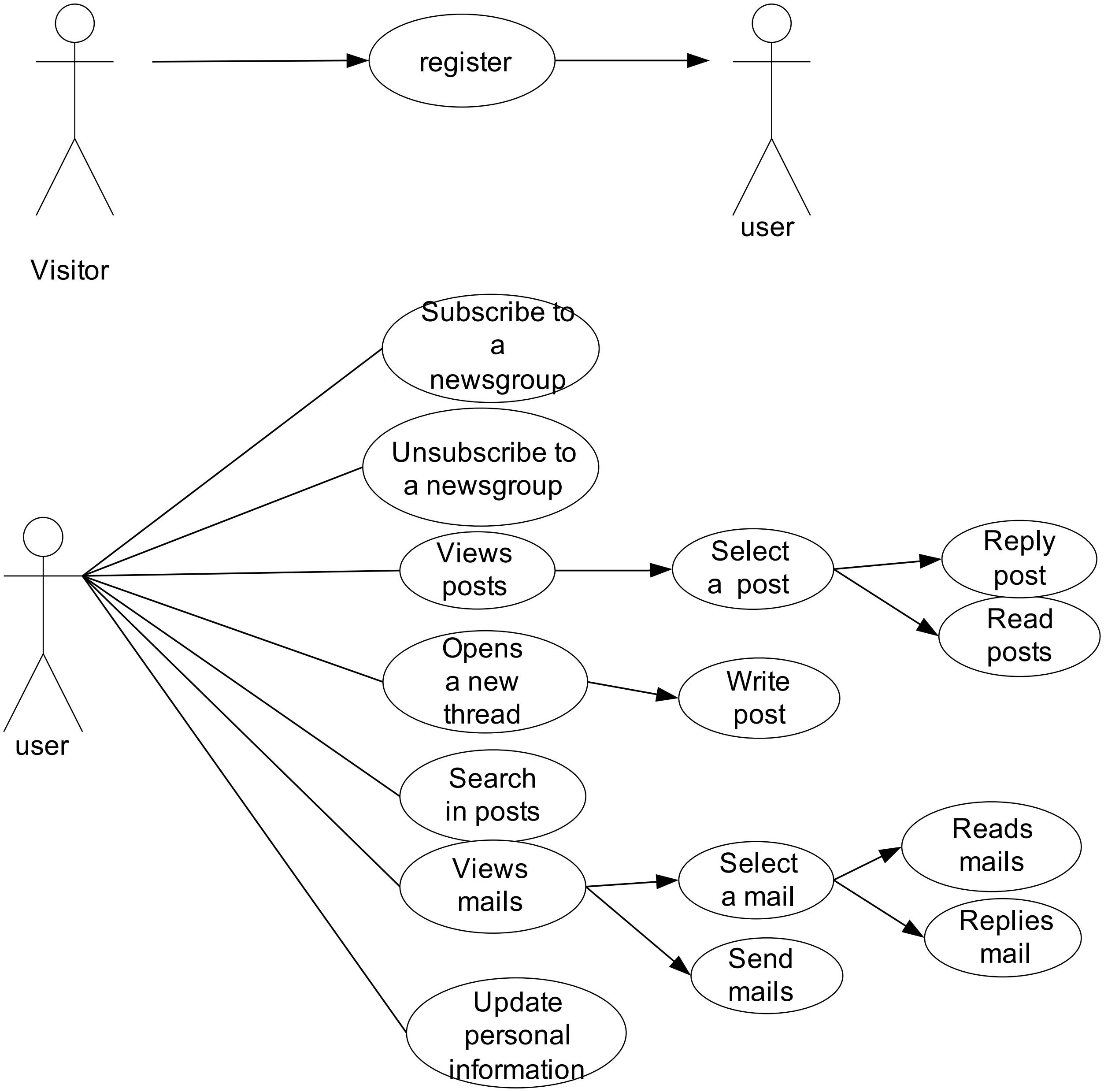
The use-cases, provide a description of how the system will be used. Once actors have been identified, use-cases can be developed. The use-case describes the manner in which an actor interacts with the system. Our project involves two main parts namely CORE and web application for USERS. Therefore, use case diagrams are also divided into two groups namely Use-Case Diagrams for CORE and Use-Case Diagrams for NewStreamLine system (web application).

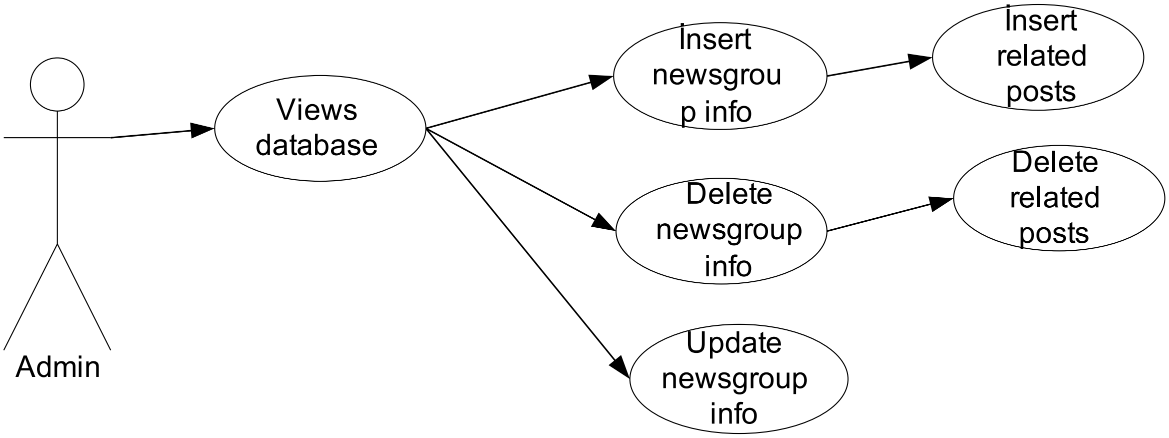
4.1.1 Use Case Diagrams for CORE

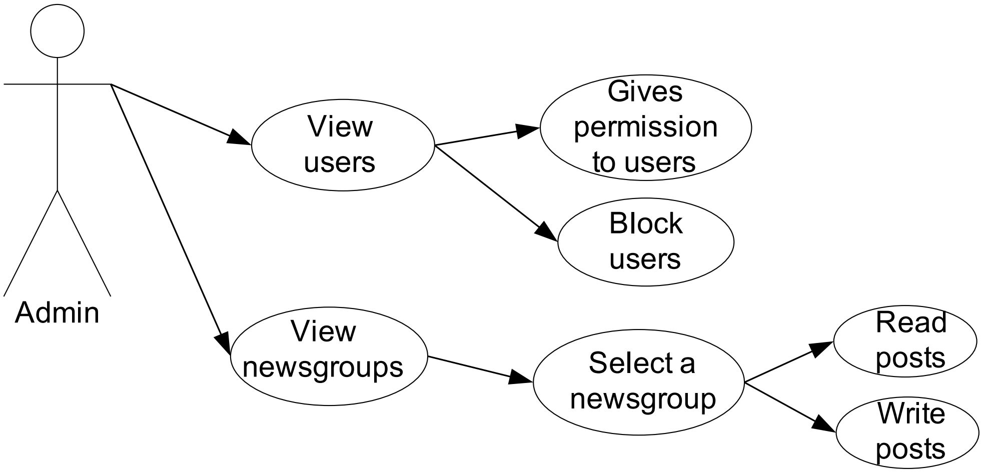
In this part, actors are servers, clients and services. In this part, actors are servers, clients and services.


CORE
Receive Mail
Send Mail
Request User Information for Mail
Reply Mail
Send Notification
Mail Server
Talk with Database Engine
Request to Send Mail
Request to Reply Mail
Request to Read Mail
Get Response from Mail Server
SMTP Service
Read Post
Request for Newsgroup Information
Write Post
Send Post
Send Notification
News Server
Request to Read & Write & Send Posts
Talk with Database Engine
Get Response from News Server
NNTP Service
SQL/XML Engine
DATABASE
Send RSS Request
RSS Client
Configure XML File
XML Service
Send User Information
Send User Requests(send mail, reply posts etc.)
Display Request Results(display posts, search results etc.)
Client
Web Applications
Response User Requests
Get User Information & Send to the Database Engine
Llisten Web Applications
Talk with Database Engine
Web Service
CORE

4.1.2 Use Case Diagrams for NewStreamLine







4.2 Functional Requirements

4.2.1 NewStreamLine System Functionality

Visitor: A visitor can use NewStreamLine system only after subscription. Therefore, visitor can only subscribe to system to be a user.

User Functionality: User must login to the system through a dialog box. User can create new topics/ create new groups/ subscribe/ unsubscribe in News Group. User can post / reply to / discuss messages. User can select articles to be sent via e-mail. User can modify account information. User can see statistics about topic reading. User can select different categories to be informed about. Lastly, user can search a topic in the system.

Admin Functionality: Admin must also log in to the system. Admin can see all newsgroups. Admin can add or delete new newsgroups to the system. Admin can view users. Admin can modify user list by blocking or giving permissions.

4.2.2 CORE Functionality

Mail Server: Mail Server always communicates with SMTP service. Mail Server can access database via SMTP service. According to requests from SMTP service, Mail Server sends, receive, reply mails. Requests occur after executing database triggers. Mail server gets the user information. At the end of the processes, Mail Server sends a notification or data to the SMTP service.

News Server: News Server always communicates with NNTP service. News Server can access database via NNTP service. According to requests from NNTP service, News Server sends, replies, writes posts. Requests occur after executing database triggers. News server gets the user information. At the end of the processes, News Server sends a notification or data to the SMTP service.

RSS Client: RSS Client send requests to configure XML file according to user preferences.

Web Applications: User can do all operation from the web application. Through web applications, user information data, user request data can be sent to web service. Web application also displays the results of these requests and data.

SMTP Service: SMTP service always waits for a change or request to be executed and to be sent to Mail Server. SMTP service always listens Database and talks with Mail Server.

NNTP Service: NNTP service always waits for a change or request to be executed and to be sent to News Server. NNTP always listens Database and talks with News Server.

XML Service: XML service responses to requests from RSS Clients.

Web Service: Web Service interacts wit the web applications. It transports the data and requests to the database engine and vice versa. It response to the web applications after executing trigger sent from database.

SQL/XML Engine: SQL/XML Engine listens all the services in the core. It puts the requests to the database in order. Every service communicates with database through this engine. Moreover, database sends all triggers and data via SQL/XML Engine.

4.3 Nonfunctional Requirements

Usability:

Our product gives service to a wide variety of users, such that people having not much knowledge about computers in profession could be able to use our system. Thus, developing a user friendly interface and usability is very important for our project developers.

Security:

Security is one of the main concerns of our product. There is a user authentication password-login system which provides a more secure and healthy working space for the users and the administrators. Moreover, the authentication is for maintaining the user hierarchy. Administrators have the right to modify the inputs and the outputs of the system, and manage to system database, modify it. Users do not have the right neither to reach to the system database, nor to modify it. One more important security aspect of our development is Secure Sockets Layer protocol, namely SSL. It is a high-level security protocol that protects the confidentiality and security of data while it is being transmitted through the internet based on RSA Data Security's public-key cryptography.

Reliability:

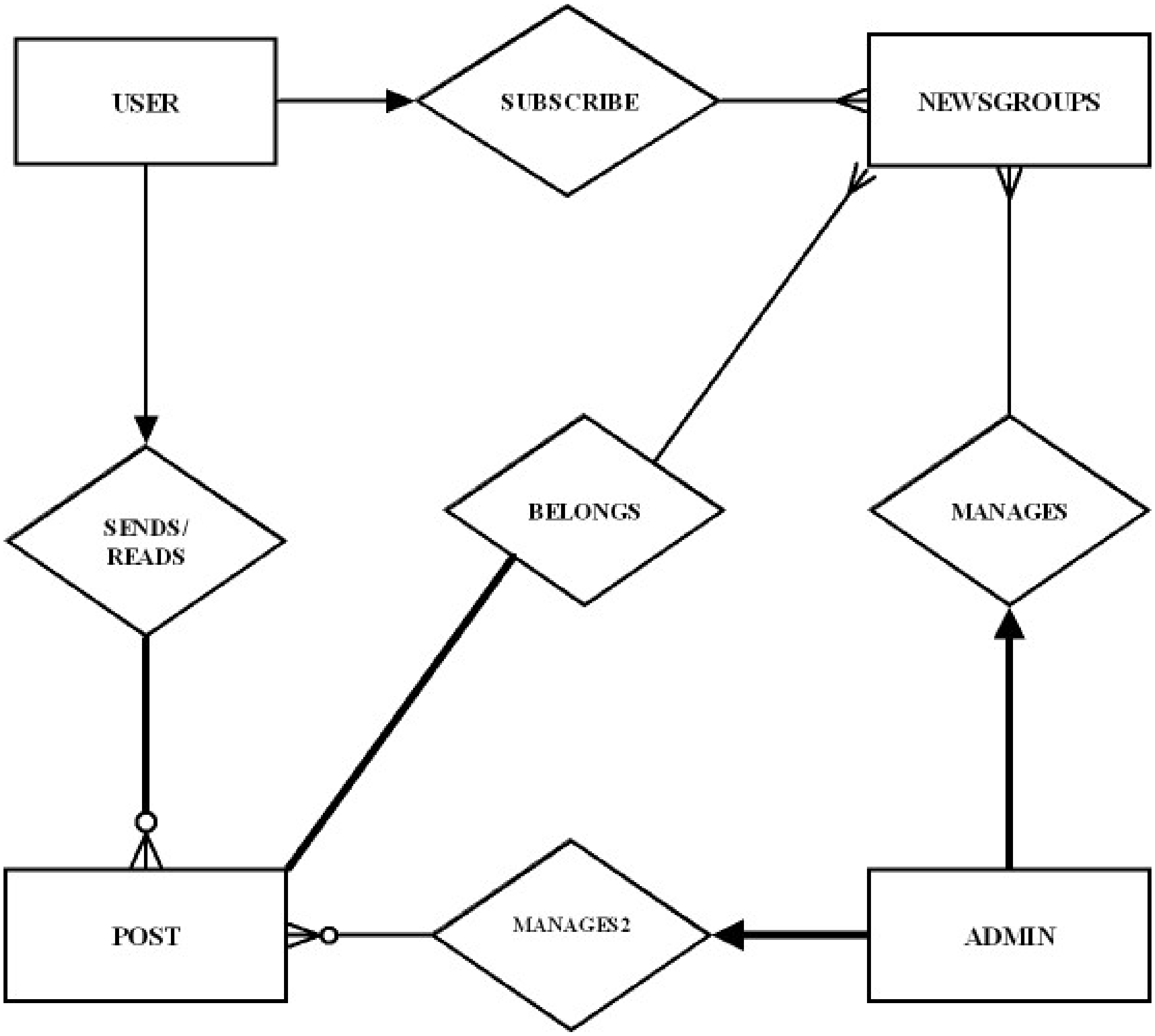
Our product will be so stable such that any minor problem will not cease the newsuni system. Moreover, we plan to do many tests after implementation to minimize the bugs on the program.

5.Modelling

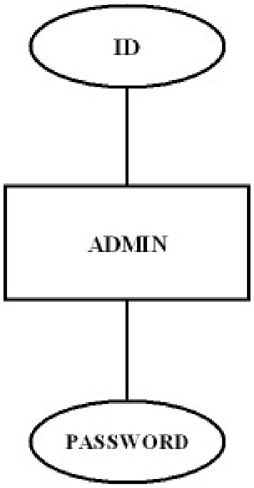
5.1 Data Model

5.1.1 ER- Diagrams

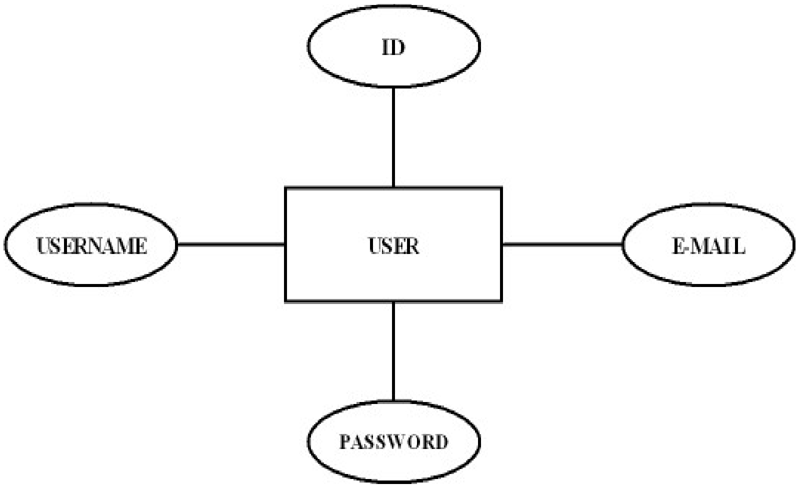
a. General View of Database



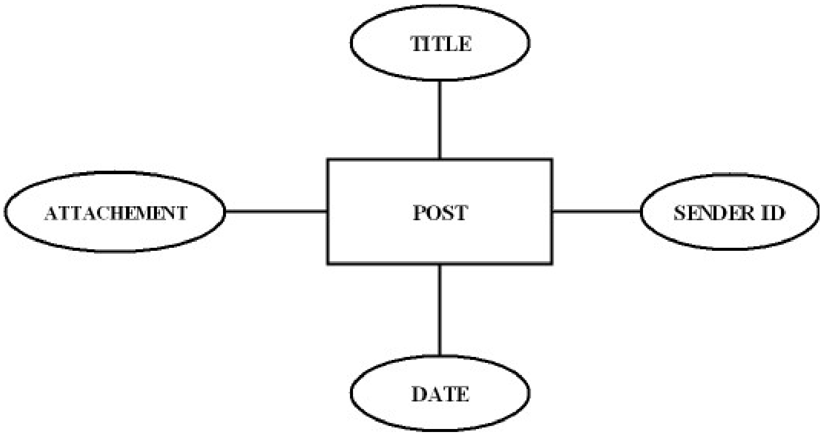
b. Admin Entity



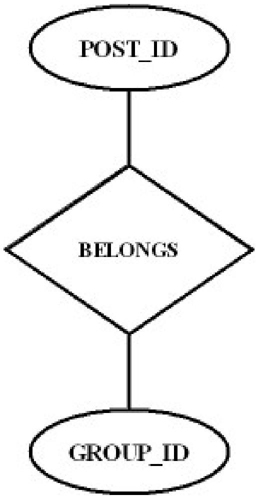
c. User Entity



d. Post Entity



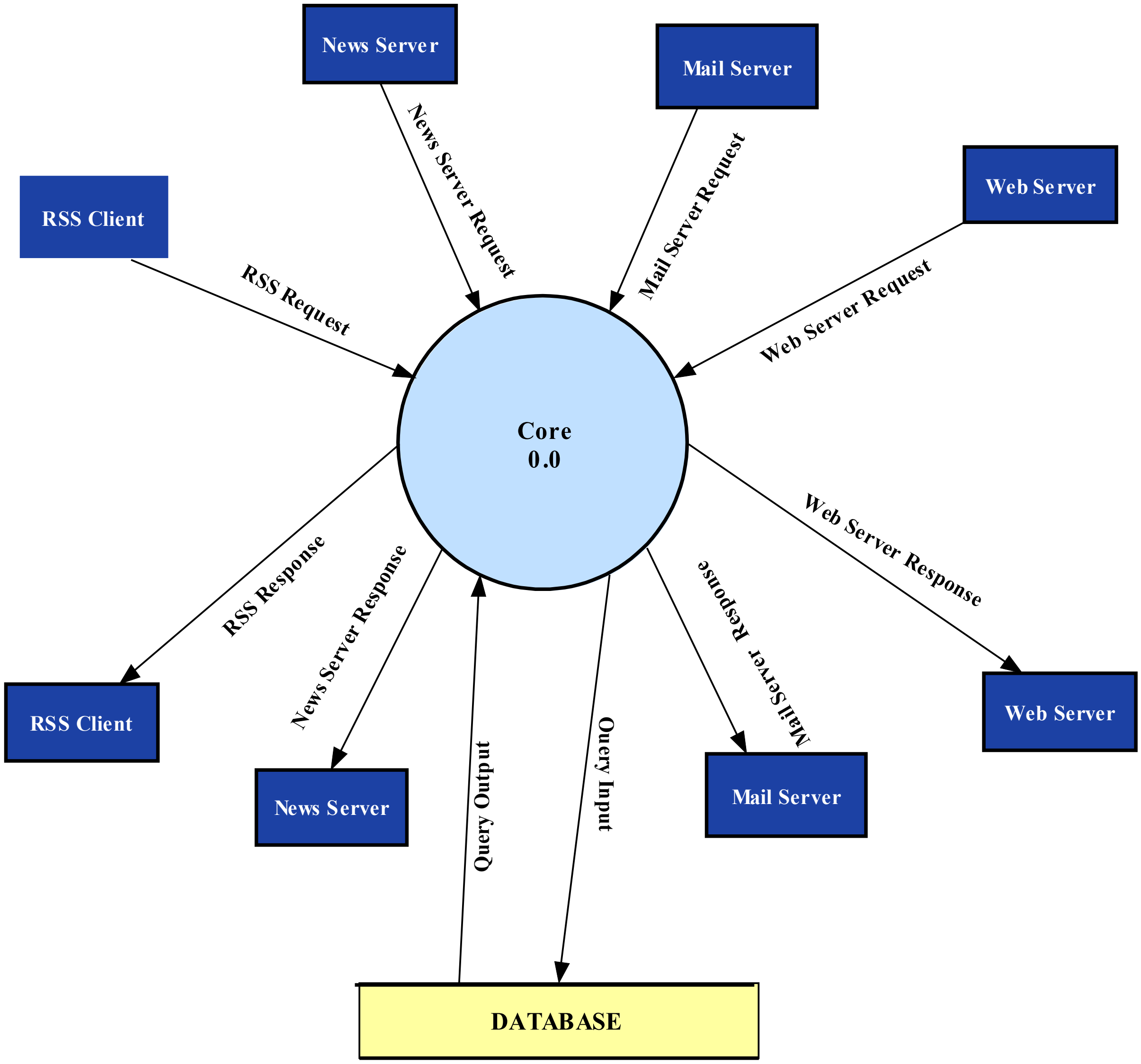
e. Belongs Relationship



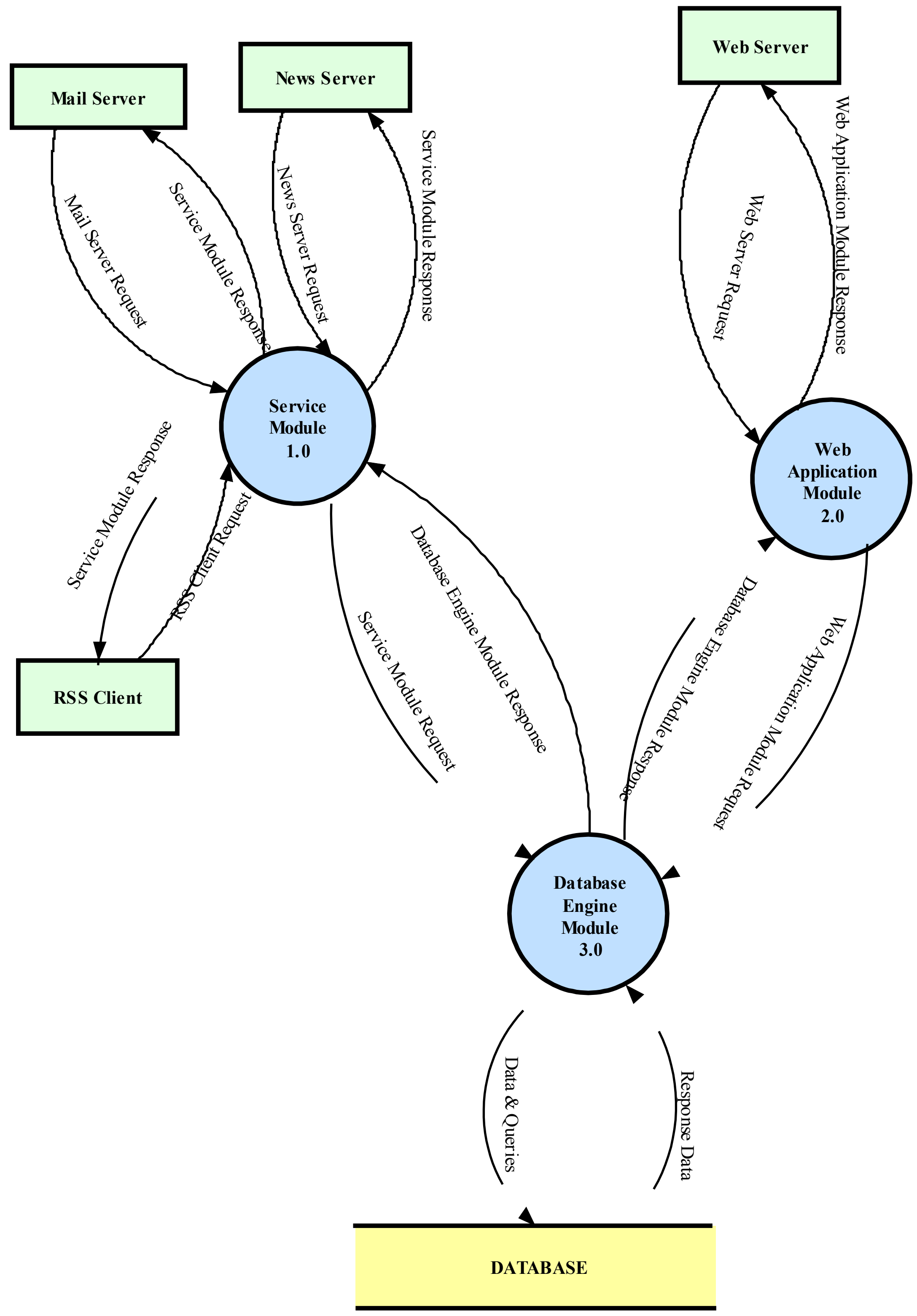
5.2 Functional Model

5.2.1 Data Flow Diagram - Core

5.2.1.1 DFD- Level 0

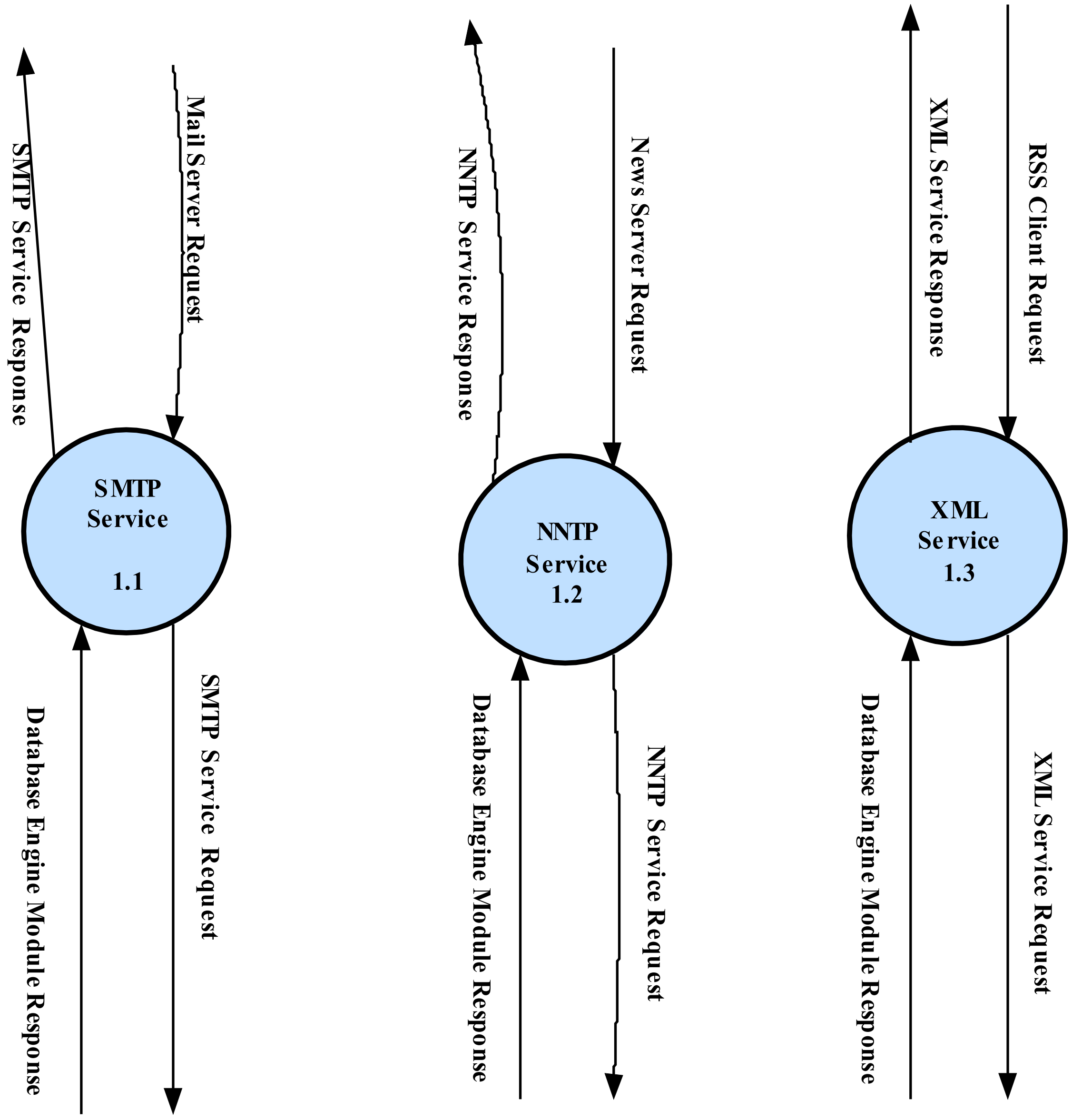


5.2.1.2 DFD- Level 1

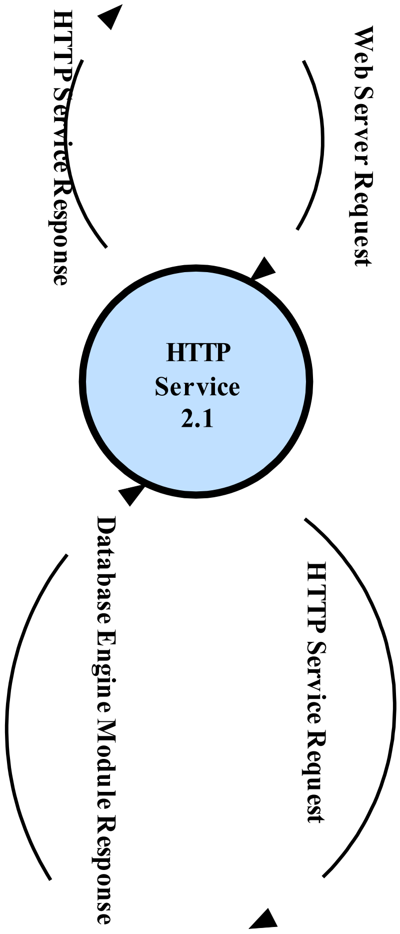


5.2.1.3 **DFD- Level 2**

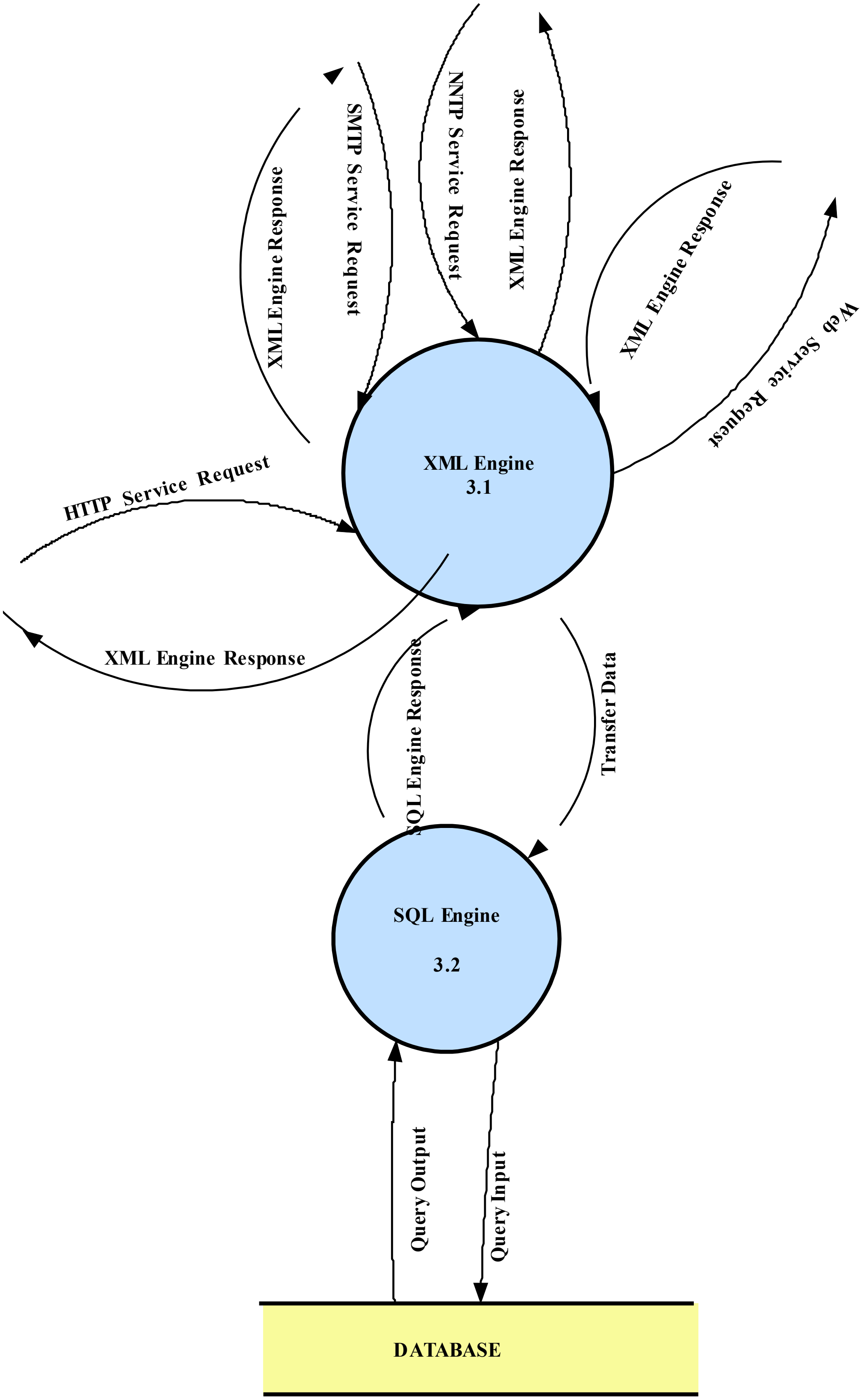
Service Module



Web Application Module

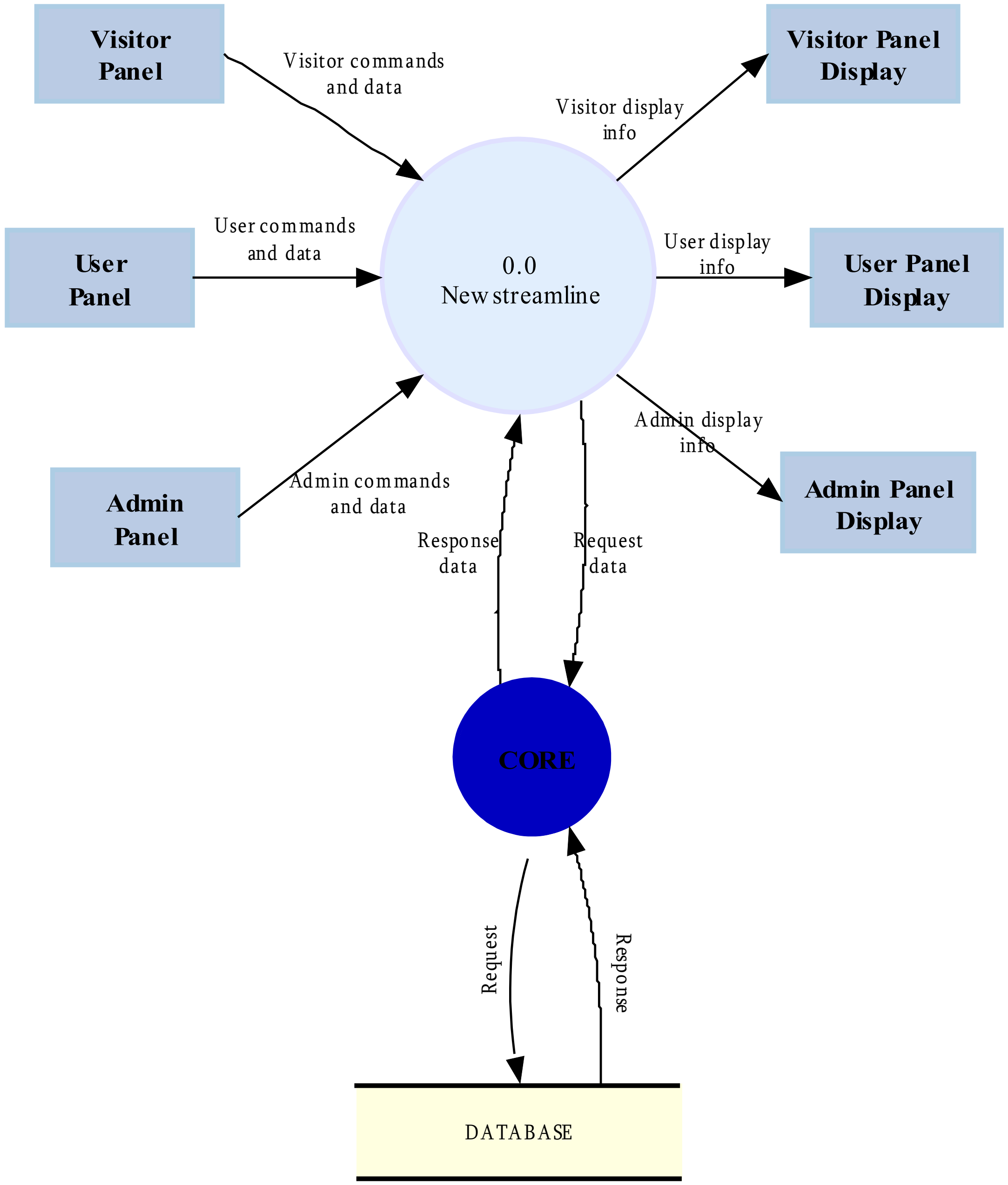


Database Engine Module

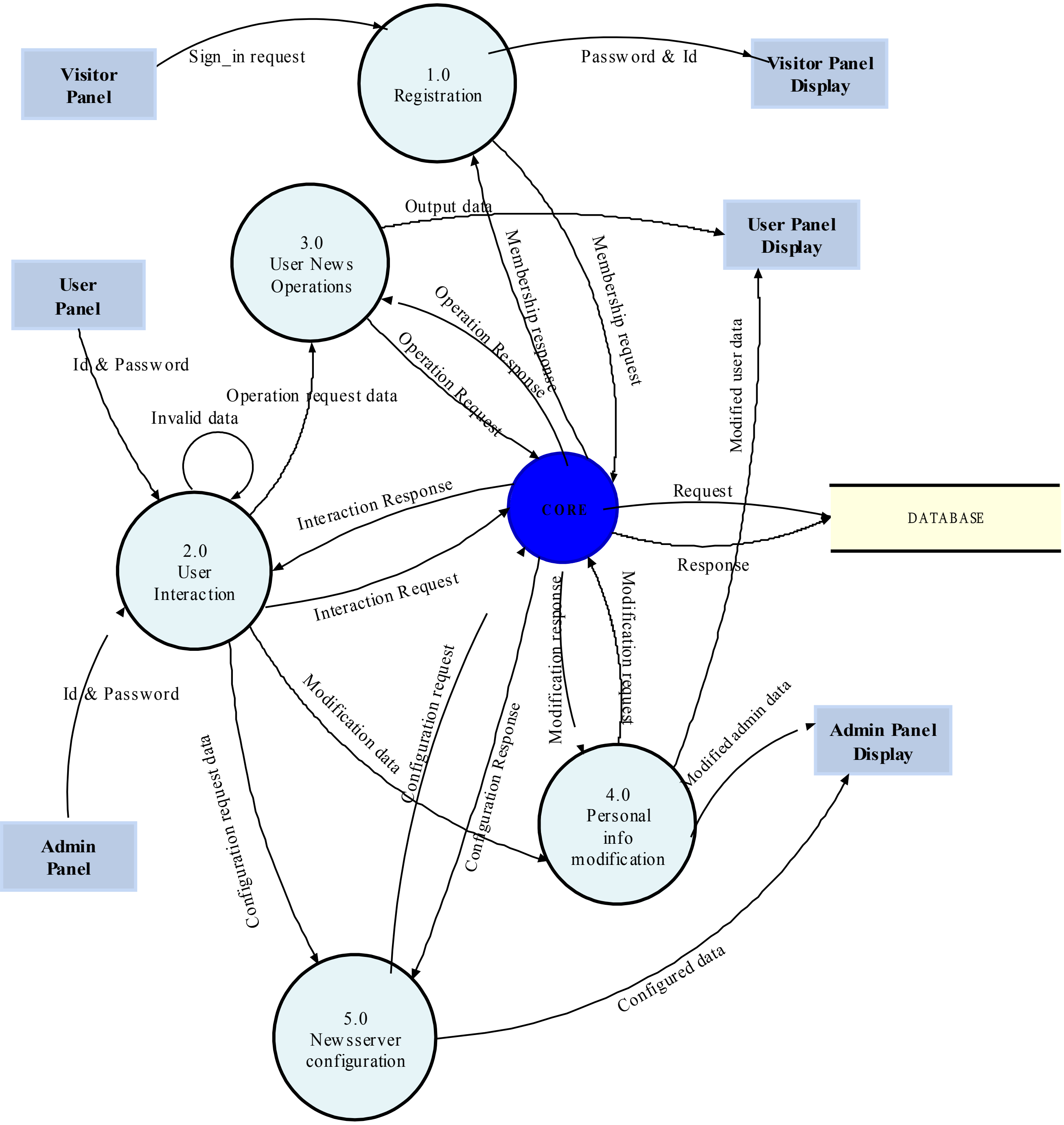


5.2.2 Data Flow Diagram – Web Application

5.2.2.1 DFD- Level 0

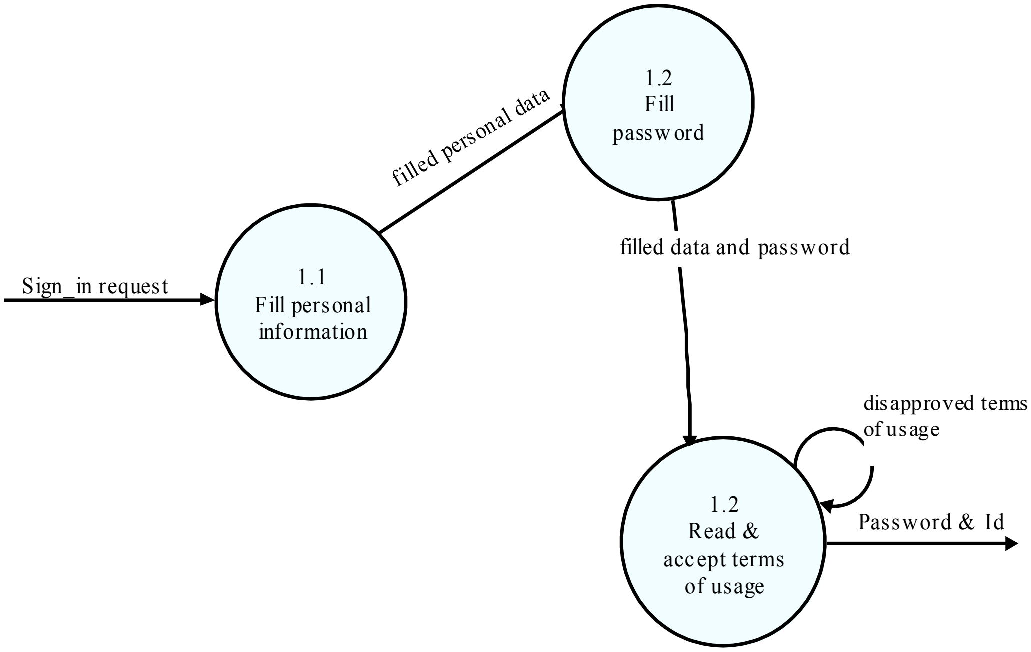


5.2.2.2 DFD- Level 1

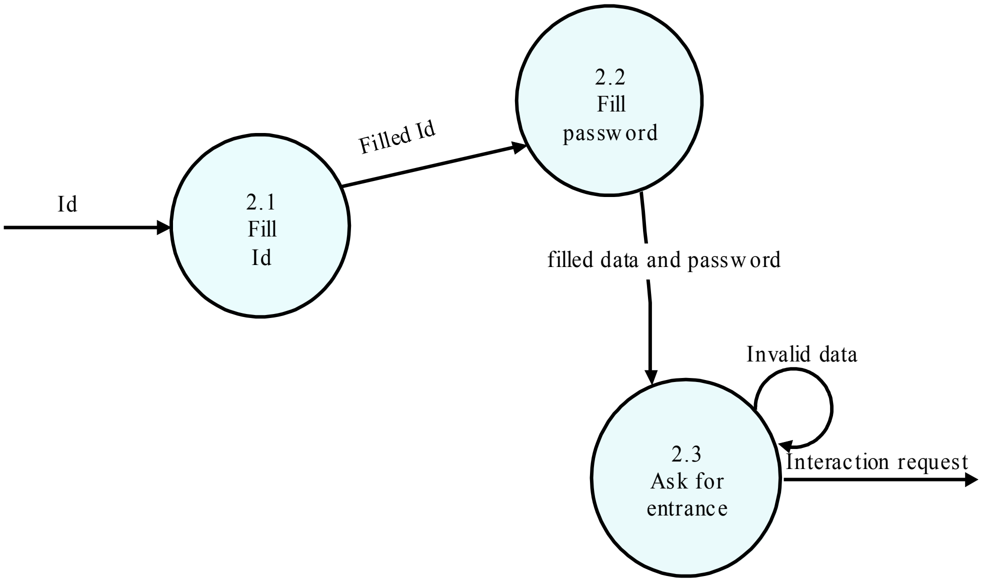


5.2.2.3 DFD- Level 2

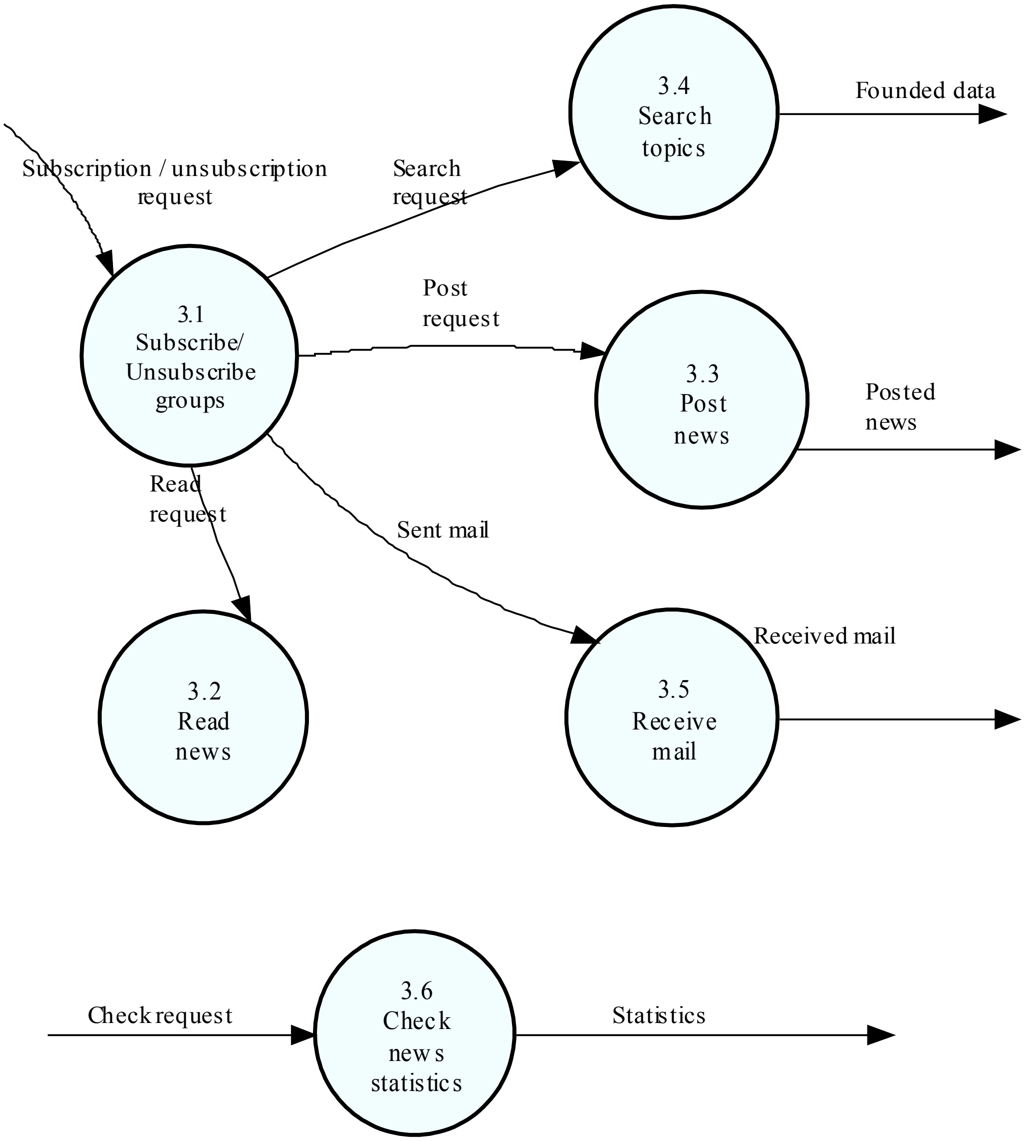
Registration Process



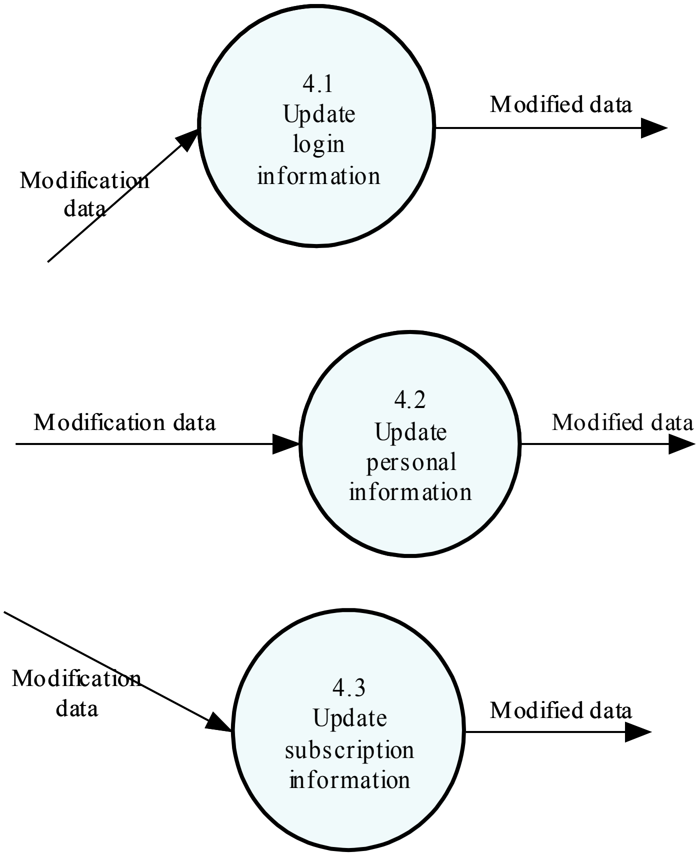
User Interaction Process



News Operations Process



Personal Info Modification Process



5.2.3 Data Dictionary:

Name:	Rss Client Request
Alias:	None
Where & How It is used:	RSS Client Input
Description:	“sended requests by RSS client ”

Name:	Rss Request
Alias:	None
Where & How It is used:	RSS Client Output
Description:	“responses to RSS client ”

Name:	News Server Request
Alias:	None
Where & How It is used:	News Server Input
Description:	“sended requests by news server ”

Name:	News Server Request
Alias:	None
Where & How It is used:	News ServerOutput
Description:	“responses to news server ”

Name:	Mail Server Request
Alias:	None
Where & How It is used:	Mail Server Input
Description:	“sended requests by mail server ”

Name:	Mail Server Request
Alias:	None
Where & How It is used:	Mail Server Output
Description:	“responses to mail server ”

Name:	Web Server Request
Alias:	None
Where & How It is used:	Web Server Input
Description:	“sended requests by web server ”

Name:	Web Server Request
Alias:	None
Where & How It is used:	Web Server Output
Description:	“responses to web server ”

Name:	Query Input
Alias:	Query request

Where & How It is used:	Accessing database
Description:	"queries from core"

Name:	Query output
Alias:	Query response
Where & How It is used:	Database output
Description:	"Output data coming from database"

Name:	Mail Server Request
Alias:	None
Where & How It is used:	Service Module Input
Description:	"Requests coming from mail server"

Name:	Service Module Response
Alias:	None
Where & How It is used:	Service Module Output
Description:	"Responses coming from service module to mails server"

Name:	News Server Request
Alias:	None
Where & How It is used:	Service Module Input
Description:	"Request coming from news server to service module"

Name:	Service Module Response
Alias:	None
Where & How It is used:	Service Module Output
Description:	"Output data coming from service module to news server"
Name:	Service Module Response
Alias:	None
Where & How It is used:	Service Module Output
Description:	"Responses coming from service module to news server"

Name:	RSS Client Request
Alias:	None
Where & How It is used:	Service Module Input
Description:	"Requests coming to service module from rss client"

Name:	Service Module Response
Alias:	None
Where & How It is used:	Service Module Output
Description:	"Responses coming from service module to rss client"

Name:	Database Engine Module Response
Alias:	None
Where & How It is used:	Database Engine Module Output
Description:	"Responses coming from database engine module "

Name:	Service Module Request
Alias:	None
Where & How It is used:	Database Engine Module Input
Description:	"Requests coming from service module to database engine module"

Name:	Service Module Request
Alias:	None
Where & How It is used:	Database Engine Module Input
Description:	"Requests coming from service module to database engine module"

Name:	Data & Queries
Alias:	None
Where & How It is used:	Database Engine Module Output
Description:	"Input data and queries coming from database engine module to database"

Name:	Response Data
Alias:	None
Where & How It is used:	Database Engine Module Input
Description:	"Input data coming from database to database engine module"

Name:	Web Application Module Request
Alias:	None
Where & How It is used:	Database Engine Module Input
Description:	"Requests coming from web application module to database engine module"

Name:	Database Engine Module Response
Alias:	None
Where & How It is used:	Database Engine Module Output
Description:	"Responses coming from database engine module to web application module"

Name:	Web Application Module Response
Alias:	None
Where & How It is used:	Web Application Module Output
Description:	"Responses coming from web application module to web server"

Name:	Web Application Module Requests
Alias:	None
Where & How It is used:	Web Application Module Input
Description:	"Requests coming from web server to web application module "

Name:	Mail Server Request
Alias:	None
Where & How It is used:	SMTP Service Input
Description:	"Requests coming to smtp service"

Name:	SMTP Service Response
Alias:	None
Where & How It is used:	SMTP Service Output
Description:	“Response coming from smtp service”

Name:	SMTP Service Request
Alias:	None
Where & How It is used:	SMTP Service Output
Description:	“Requests coming from smtp service

Name:	Database Engine Module Response
Alias:	None
Where & How It is used:	SMTP Service Input
Description:	“Responses coming to smtp service

Name:	News Server Request
Alias:	None
Where & How It is used:	NNTP Service Input
Description:	“Requests coming to nntp service”

Name:	NNTP Service Response
Alias:	None
Where & How It is used:	NNTP Service Output
Description:	“Response coming from nntp service”

Name:	NNTP Service Request
Alias:	None
Where & How It is used:	NNTP Service Output
Description:	“Requests coming from nntp service

Name:	Database Engine Module Response
Alias:	None
Where & How It is used:	NNTP Service Input
Description:	“Responses coming to nntp service

Name:	Rss Client Request
Alias:	None
Where & How It is used:	XML Service Input
Description:	“Requests coming to xml service”

Name:	XMLService Response
Alias:	None
Where & How It is used:	XMLService Output
Description:	“Response coming from xml service”

Name:	XML Service Request
Alias:	None
Where & How It is used:	XML Service Output
Description:	“Requests coming from xml service

Name:	Database Engine Module Response
Alias:	None
Where & How It is used:	XML Service Input
Description:	“Responses coming to xml service

Name:	Web Server Request
Alias:	None
Where & How It is used:	HTTP Service Input
Description:	“Requests coming to http service”

Name:	HTTP Service Response
Alias:	None
Where & How It is used:	HTTP Service Output
Description:	“Response coming from http service”

Name:	HTTP Service Request
Alias:	None
Where & How It is used:	HTTP Service Output
Description:	“Requests coming from http service

Name:	Database Engine Module Response
Alias:	None
Where & How It is used:	HTTP Service Input
Description:	“Responses coming to http service

Name:	Visitor commands and data
Alias:	None
Where & How it is used:	Visitor Input
Description:	“written input data and interface commands by visitor”

Name:	User commands and data
Alias:	None
Where & How it is used:	User Input
Description:	“written input data and interface commands by user”

Name:	Admin commands and data
Alias:	None
Where & How it is used:	Administrator Input
Description:	“written input data and interface commands by admin”

Name:	Visitor commands and data
Alias:	None
Where & How it is used:	Visitor Input
Description:	“written input data and interface commands by user”

Name:	Visitor display info
Alias:	None
Where & How it is used:	Output of the *Newstreamline* News system
Description:	“output of the system which is shown on the visitor monitor”

Name:	User display info
Alias:	None
Where & How it is used:	Output of the *Newstreamline* News system on t
Description:	“output of the system which is shown on the admin monitor”

Name:	Admin display info
Alias:	None
Where & How it is used:	Output of the *Newstreamline* News system
Description:	“output of the system which is shown on the admin monitor”

Name:	Response data
Alias:	Response
Where & How it is used:	Output of the core system to the responding requests & Input of the *Newstreamline* system
Description:	“output of the system which is shown on the visitor monitor”

Name:	Request data
Alias:	Request
Where & How it is used:	Input of *Core* according to the *Newstreamline* system
Description:	“output of the system which is shown on the visitor monitor”

Name:	Sign_in request
Alias:	None
Where & How it is used:	Input of Registration(1.0)
Description:	“it is the desired input coming from visitor panel to register to the system”

Name:	Password & Id

Alias:	Visitor commands output data
Where & How it is used:	Output of the Registration(1.0)
Description:	“given password & id to the visitor, it is a display message of the Visitor Display Panel ”

Name:	Membership request
Alias:	Request data
Where & How it is used:	Input of *Core* according to the Registration(1.0)
Description:	“It is an request for being a member of the system”

Name:	Membership response
Alias:	Response data
Where & How it is used:	Output of *Core*
Description:	“It is a response to the system for being a member of the system”

Name:	Id & Password
Alias:	None
Where & How it is used:	Input of User Interaction(2.0)
Description:	“the id and password of user/admin given by the system in order to login”

Name:	Invalid data
Alias:	None
Where & How it is used:	Output of the User Interaction(2.0)
Description:	“an error message that shows the invalidation of Id and password, it does not let the user/admin log in”

Name:	Interaction Request
Alias:	Request data
Where & How it is used:	Input of *Core* according to the User Interaction(2.0)
Description:	“It is an request for being a member of the system”

Name:	Interaction Response
Alias:	Response data
Where & How it is used:	Output of *Core*
Description:	“it is a response to the system for logging in to the system”

Name:	Operation request data
Alias:	None
Where & How it is used:	Input of User News Operations(3.0)
Description:	"neccassary input to operate on news system"

Name:	Operation Request
Alias:	Request data
Where & How it is used:	Input of *Core* according to the News Operations(3.0)
Description:	"it is an request for doing news operations"

Name:	Operation Response
Alias:	Response data
Where & How it is used:	Output of *Core*
Description:	"it is a response to the system doing news operations"

Name:	Output data
Alias:	User display info
Where & How it is used:	Output of User News Operations(3.0)
Description:	"it is a response to the system for logging in to the system"

Name:	Modification data
Alias:	None
Where & How it is used:	Input to the Personal info modification(4.0)
Description:	"it is neccessary data to modify personal information"

Name:	Modification Request
Alias:	Request data
Where & How it is used:	Input of *Core* according to the Personal info modification(4.0)
Description:	"it is an request for updating personal information"

Name:	Modificaiton Response
Alias:	Response data
Where & How it is used:	Output of *Core*
Description:	"it is a response to the system for updating personal information"

Name:	Modified user data
Alias:	User display info

Where & How it is used:	Output of the Personal info modification(4.0)
Description:	"it is a display message according to the modification of the user data"

Name:	Modified admin data
Alias:	Admin display info
Where & How it is used:	Output of the Personal info modification(4.0)
Description:	"it is a display message according to the modification of the admin data"

Name:	Configuration request data
Alias:	None
Where & How it is used:	Input of the Newsserver configuration(5.0)
Description:	"it is neccessary data to configure the system"

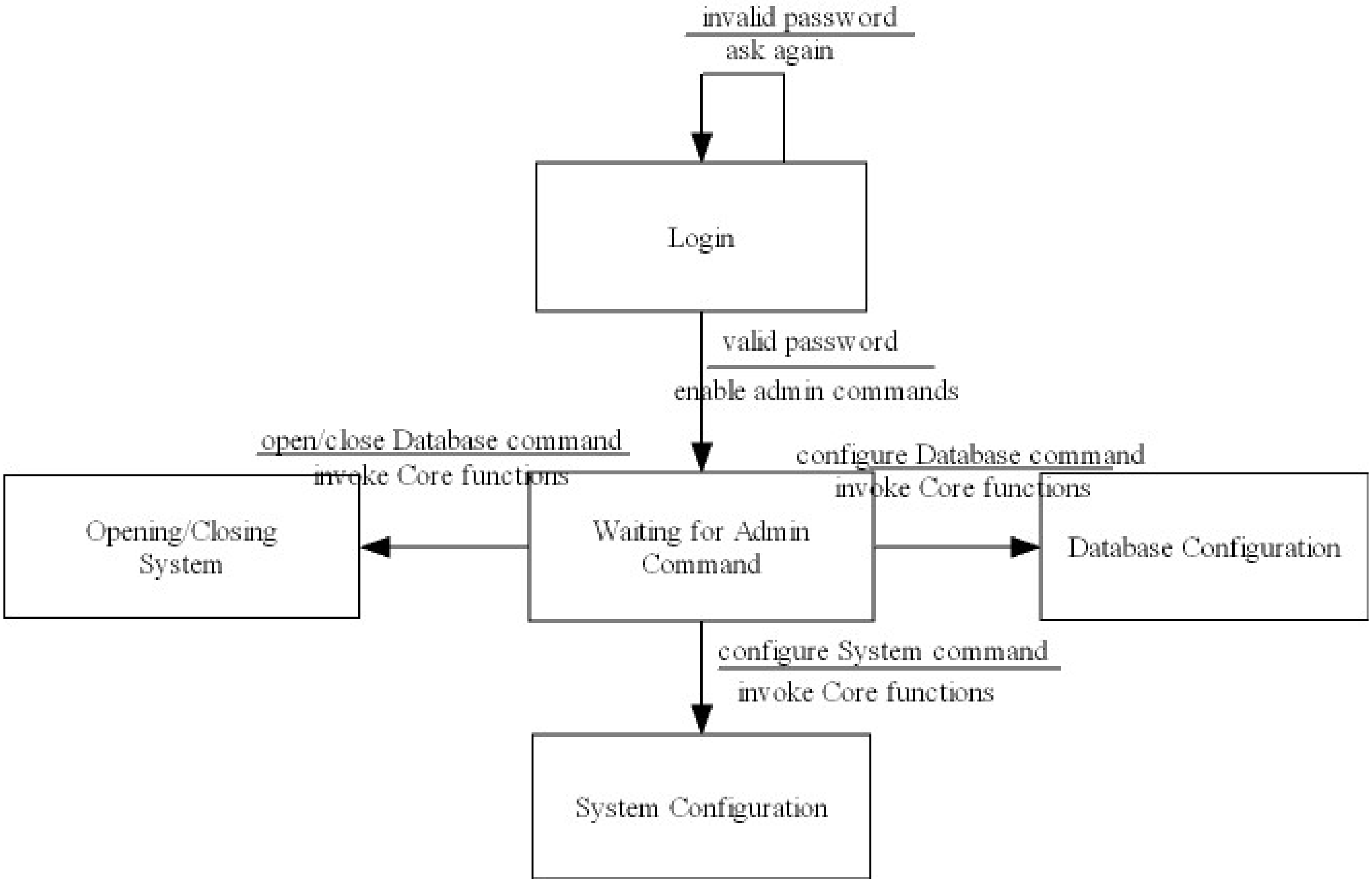
Name:	Configuration Request
Alias:	Request data
Where & How it is used:	Input of *Core* according to the Newsserver configuration(5.0)
Description:	"it is an request for configuring the system"

Name:	Configuration Response
Alias:	Response data
Where & How it is used:	Output of *Core*
Description:	"it is a response to the system for configuring the system"

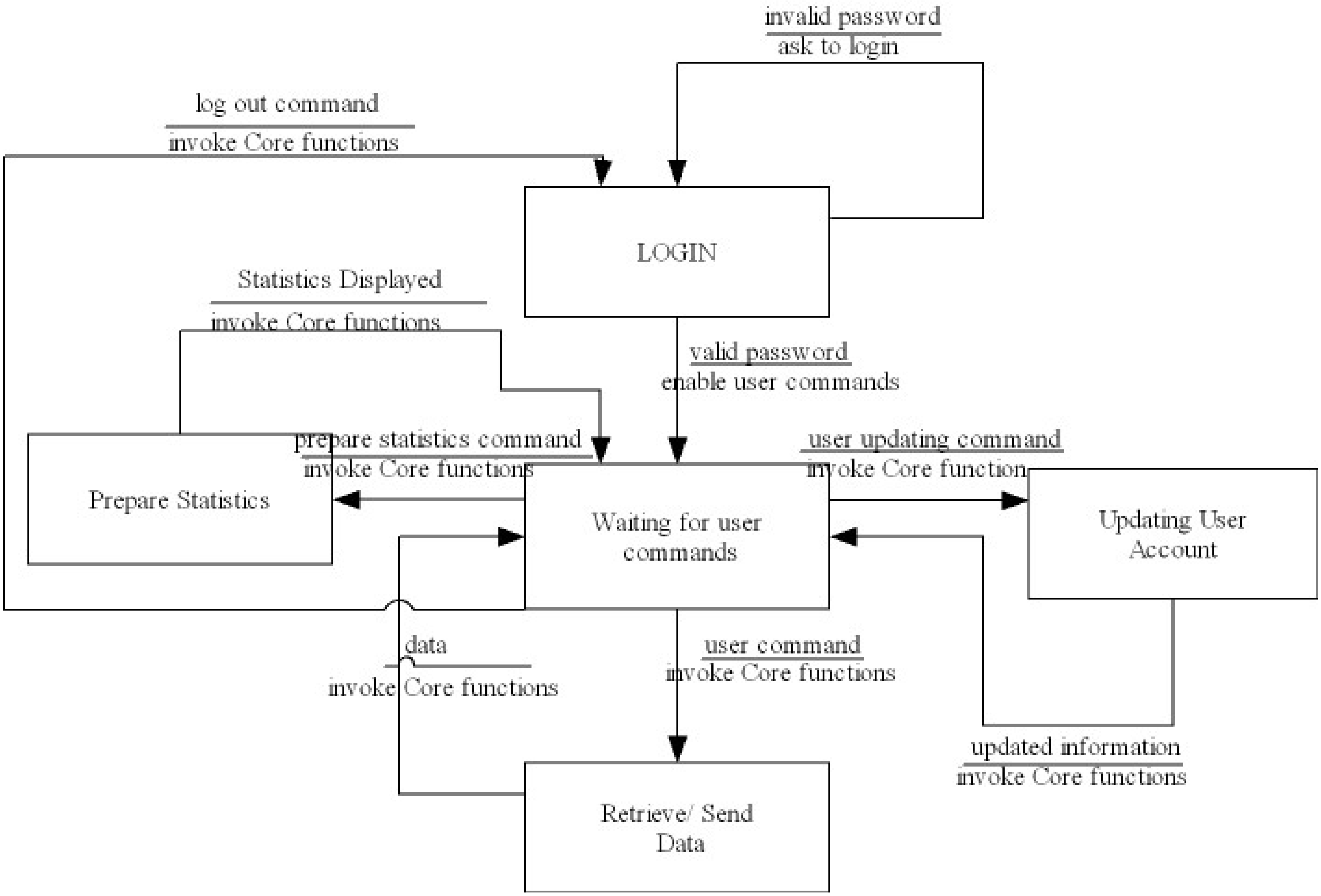
Name:	Configuration data
Alias:	Admin display info
Where & How it is used:	Output of the Newsserver configuration(5.0)
Description:	"it is an admin display message according to the configuration of the system"

5.3 Behaviour Model

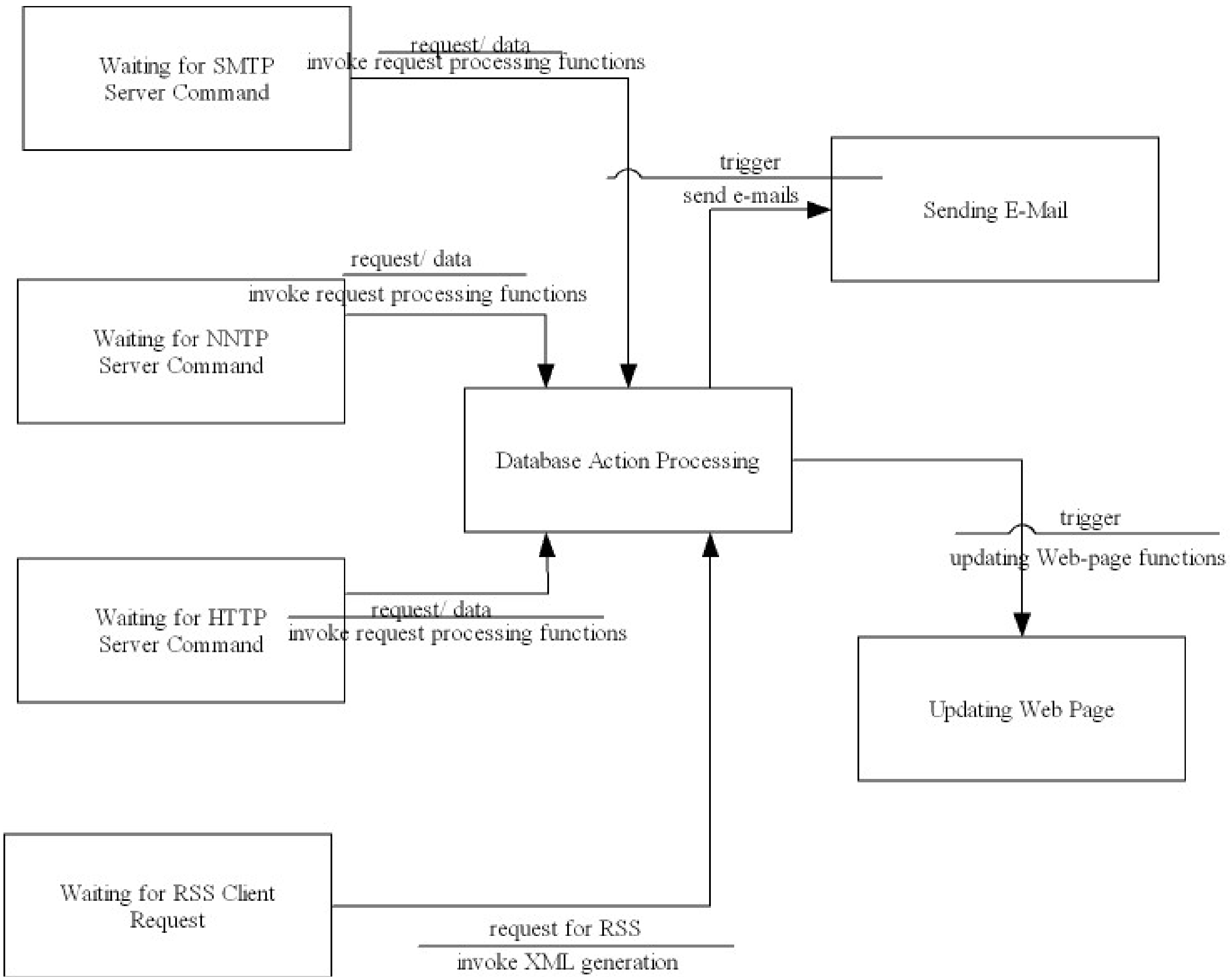
5.3.1 State Transition Diagram for Admin



5.3.2 State Transition Diagram for User



5.3.3 State Transition Diagram for Core


Waiting for SMTP Server Command
request/ data
invoke request processing functions
trigger
send e-mails
Sending E-Mail
Waiting for NNTP Server Command
request/ data
invoke request processing functions
Database Action Processing
trigger
updating Web-page functions
Waiting for HTTP Server Command
request/ data
invoke request processing functions
Updating Web Page
Waiting for RSS Client Request
request for RSS
invoke XML generation

6 Market Research

6.1 Literature Survey:

Before starting to this project, our knowledge about the architecture and the working mechanism of news servers was restricted. In order to gain the sufficent background, to ensure that the project is realizable and to broaden our imaginations, the market research was a must. We have made an extensive area research and examined the software products in detail that has been produced so far. We researched the existing news servers as Freshmeat, WebLogo and dBase. We tried to solve their working mechanisms and principles, which features are present, or mostly absent and desired. Now we have a satisfying knowledge to develop and release a successful product to the software market.

From the market search, we have seen that there is not an exact example of our product; but only there are some applications that resembles to newsuni. Companies with small budgets do not use even a basic news software, they prefer using emails instead of a resident news server system. Moreover, big companies or establishments like the Central Bank of Turkiye Cumhuriyeti are pleased with using such applications, however we had no chance to get information about their system since the security in such places is in high level.

In order to deepen our research, we divided our subject “news server” into 5 subtopics, namely SMTP, NNTP, RSS, HTTP and Web Servers. These concepts will play important roles in our project. There is a summation part below which gives some basic information about each subtopic one by one:

SMTP:

SMTP is the short form of Simple Mail Transfer Protocol, a protocol for sending e-mail messages between servers. Most e-mail systems that send mail over the Internet use SMTP to send messages from one server to another; the messages can then be retrieved with an e-mail client using either POP or IMAP. In addition, SMTP is generally used to send messages from a mail client to a mail server. This is why you need to specify both the POP or IMAP server and the SMTP server when you configure your e-mail application. In our Project SMTP server will be used in e-mailing system.

NNTP:

Network News Transfer Protocol, namely NNTP, is a protocol that gives the ability to distribute news article, inquire and retrieve them using a reliable stream client-server model. There is a database where all news topics are stored. Database is built on the server machine and other machines which are on the LAN or the members of cluster connect to it and the data exchange acquires. This is done by defined commands which are coded as a three digit numbers. Also there are codes for Responses. There are different ways in which articles travel between computers. It can be done only in one or in all directions. Different mechanisms can be defined to prevent the same message to be sent to the same client.

RSS:

RSS is a format for syndicating news and the content of news-like sites, including major news sites like BBC, news-oriented community sites like Slashdot, and personal weblogs. It is a simple XML-based system that allows users to subscribe to their favorite websites. RSS offers access to any kind of frequently posted information. This also makes it an interesting technology for handling usenet newsgroups, which are like updated several times a day. Since RSS allows newsgroups to distribute news, announcements, and blog posts directly to end customers, in our project we plan to use RSS feed to distribute newsgroups' news and announcements.

HTTP:

HTTP is the abbreviation of Hypertext Transfer Protocol. HTTP is the network protocol of the Web. It is used to transmit resources. HTTP uses the client-server model: An HTTP client opens a connection and sends a request message to an HTTP server; the server then returns a response message, usually containing the resource that was requested. Our communication with the web can be accomplished according to HTTP.

Web Server:

A Web server is a computer that is responsible for accepting HTTP requests from clients through Web browsers, and serving them HTTP responses along with optional data contents, which are usually Web pages such as HTML documents and linked objects. Most of the modern web servers firstly resolve the name of the Web page requested, then authenticate the client, perform access control on the client and on the Web page, check the cache and fetch the requested page from disk. Then determine the MIME (Multipurpose Internet Extensions) type to include in the response and take care of miscellaneous odds and ends. Finally return the reply to the client and make an entry in the server log. The usability of web servers differs due to their performance, main features, load limits, path translation, and overload symptoms. It is known that the most popular server is the Apache Web Server.

6.2 Interviews

Meeting with Cevat Şener:

We have made several interviews to make the project topic clear. One of these interviews is done with Dr.Cevat Şener. Firstly, we introduce our project topic to him. He explained us the overall view of the architecture and the modular view of the project. He suggested us to use Java progrmming language since it is platform independent.Then, he adviced us to search about sockets and ports.

Meeting with Nihan Kesim Çiçekli:

In a group of 3 people we went to Assoc.Prof.Dr. Nihan Kesim Çiçekli and talk about databases with her, as she teaches Ceng351 and Ceng352 courses. She gave us some information about indexing, and explained differences between Oracle Database and MYSQL. Because of that explanation we decided to choose MYSQL for our project. Moreover, she gave as some book references about three-tired computing and suggested to search deeply about middleware.

Meeting with Nihan Kesim Çiçekli was our first research in these areas and she helped us a lot to understand the project.

Meeting with Pinar Şenkul:

We consulted Pinar Senkul some aspects about our database system. She informed us about the main differences of Oracle and MySQL Database Management Systems. After this successful meeting, we decided to MySQL DBMS for our project's database backend.

Interview with Büşra Çekkayan

As an extent of market research, we wanted to arrange a meeting with a computer engineer who is working on the market place. Mrs.Çekkayan was one of our friend's supervisor engineer in summer practice. Thus, we have not faced with any difficulties with the arrangement. It was a 30 minutes interview. We have discussed our database, our project programming language, and modules of the Project. At the end of meeting we have concluded below:

- Use JAVA as programming language
- Use MySQL as database.
- Use JAVA version control during implementation of the code.

7. USER SCENARIO

Communication is the base relation between people. It is a fact that Newstreamline is very useful tool for communication. To clarify you our system, we have prepared a user scenario related to communication.

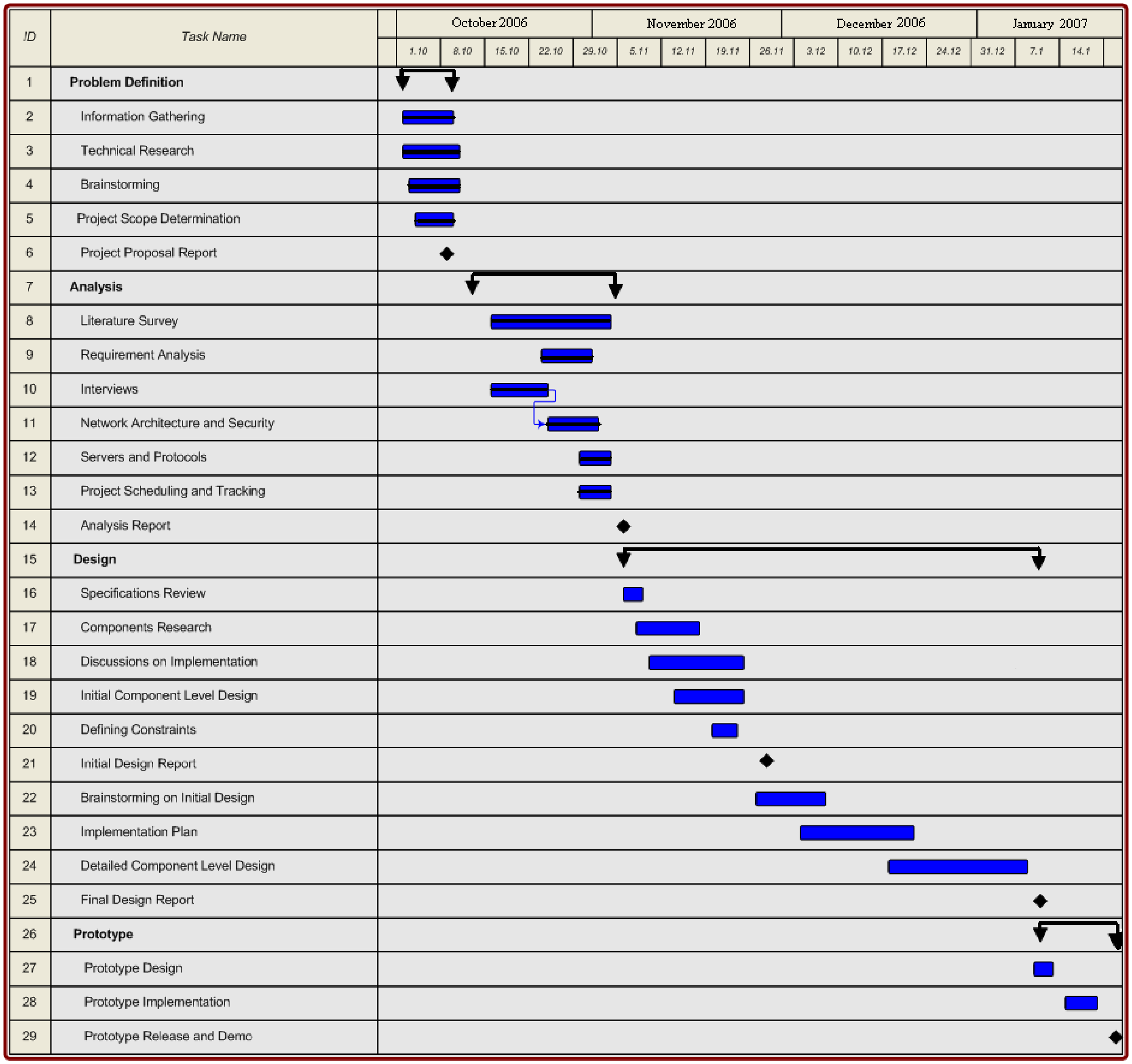
Ahmet is a new employee in the NewStreamLine Company. He is also new to his project area. He does not know anyone at his work place. At the beginning, he feels himself alone, and can not easily ask anything to his colleagues. One day, he has an important problem with his project, and he goes asking one of the colleagues from his project. However, the colleague has an important meeting and has to be there in time. Before he leaves the company, he suggests him to us the NewStreamLine system. He said that he can find a solution to his problem there very easily. Then, Ahmet decides to try this system. After the registration process is finished, he finds the related group to his problem just after viewing the news groups list. After that he sends a post about his problem to this group and immediately gets a reply which suggests a smart solution. Furthermore, as the time passes, he gets more replies to his post. He solves his problem easily by newstreamline news system. While he is viewing the system, he notices that there is a search engine which provides related posts searching criteria like according to topic of post, date or author. Also he finds a part where a user can get statistics about. By this tool, now he can learn most viewed news.

Ahmet wants to go to cinema. However, he is new to the news server system and everything is confusing to him. He logs into the Newstreamline News Server System and runs an eye over the announcements page. He also finds a friend from the friends news group. They go to the cinema together and have a very good weekend.

Finally, he likes the system very much and thankful to his colleague due to his advice.

8. APPENDIX

Gannt Chart



ID	Task Name
1	**Problem Definition**
2	Information Gathering
3	Technical Research
4	Brainstorming
5	Project Scope Determination
6	Project Proposal Report
7	**Analysis**
8	Literature Survey
9	Requirement Analysis
10	Interviews
11	Network Architecture and Security
12	Servers and Protocols
13	Project Scheduling and Tracking
14	Analysis Report
15	**Design**
16	Specifications Review
17	Components Research
18	Discussions on Implementation
19	Initial Component Level Design
20	Defining Constraints
21	Initial Design Report
22	Brainstorming on Initial Design
23	Implementation Plan
24	Detailed Component Level Design
25	Final Design Report
26	**Prototype**
27	Prototype Design
28	Prototype Implementation
29	Prototype Release and Demo