Software Requirements Specification

for

Information System for Managing Town Hall Activities

Version 1.2

Prepared by Stănescu Valentin Mihai
Deaconescu Andrei Cornel
Cojocaru Ionuţ Cosmin

12.03.2017

Table of Contents

Table of Contents 1
Revision History 1
1. Introduction 2
1.1 Purpose 2
1.2 Product Scope 2
1.3 Intended Audience and Reading Suggestions 2
1.4 Definitions, Acronyms, and Abbreviations 2
1.5 References 2
2. Overall Description 2
2.1 Product Perspective 2
2.2 Product Functions 2
2.3 User Classes and Characteristics 4
2.4 Operating Environment 4
2.5 Design and Implementation Constraints 4
2.6 User Documentation 4
2.7 Assumptions and Dependencies 5
3. External Interface Requirements 5
3.1 User Interfaces 5
3.2 Hardware Interfaces 5
3.3 Software Interfaces 5
3.4 Communications Interfaces 6
4. System Features 6
5. Other Nonfunctional Requirements 9
5.1 Performance Requirements 9
5.2 Safety Requirements 9
5.3 Security Requirements 9
5.4 Software Quality Attributes 9
5.5 Business Rules 10
6. Other Requirements 10
Appendix A: Glossary 10
Appendix B: Analysis Models 10
Appendix C: To Be Determined List 10

Revision History

Name	Date	Comments	Version
Stănescu Valentin Mihai Deaconescu Andrei Cornel Cojocaru Ionuţ Cosmin	12.03.2017	First Revision	1.0
Stănescu Valentin Mihai Deaconescu Andrei Cornel Cojocaru Ionuţ Cosmin	22.03.2017	Use case diagram, activity diagram	1.1
Stănescu Valentin Mihai Deaconescu Andrei Cornel Cojocaru Ionuţ Cosmin	29.03.2017	Added more functional req. Class diagram	1.2

1. Introduction

1.1 Purpose

The purpose of this document is to present a detailed description of the Information System for managing Town Hall activities application. It will explain the features of the system, the interfaces of the system and what the system will do.

1.2 Product Scope

The system will be built on a client/server architecture for assisting the mayor on achieving the following tasks:

- Establishing the municipality's organizational chart
- Communicating with the citizens through online audiences
- Document flow management within the municipality

1.3 Intended Audience and Reading Suggestions

This document is intended for both the municipality and the developers of the system.

1.4 Definitions, Acronyms, and Abbreviations

The **organizational chart** represents the details of the municipality's organization, subdivisions and the relations between the employees and the departments of the institution.
The **online audience** is an online meeting between a user and a municipality employee.
The **document flow management** is an informatics system that allows the distribution, storage and finding the documents in electronic format.

1.5 References

GitHub project link: https://github.com/Valentin-Mihai/InformationSystemsProject

2. Overall Description

2.1 Product Perspective

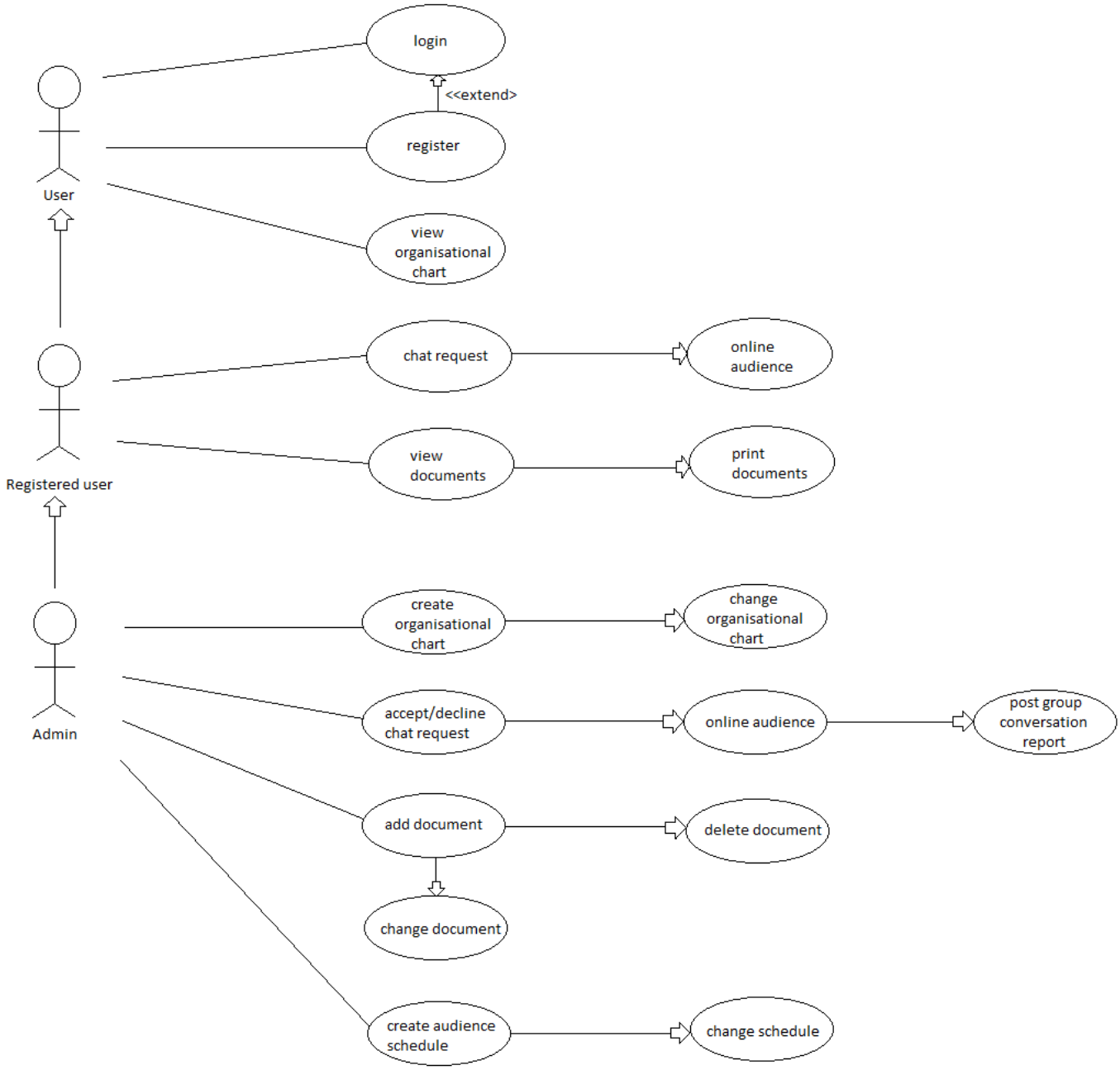
This product is a self-contained system intended for managing a variety of activities inside a town hall environment.

2.2 Product Functions

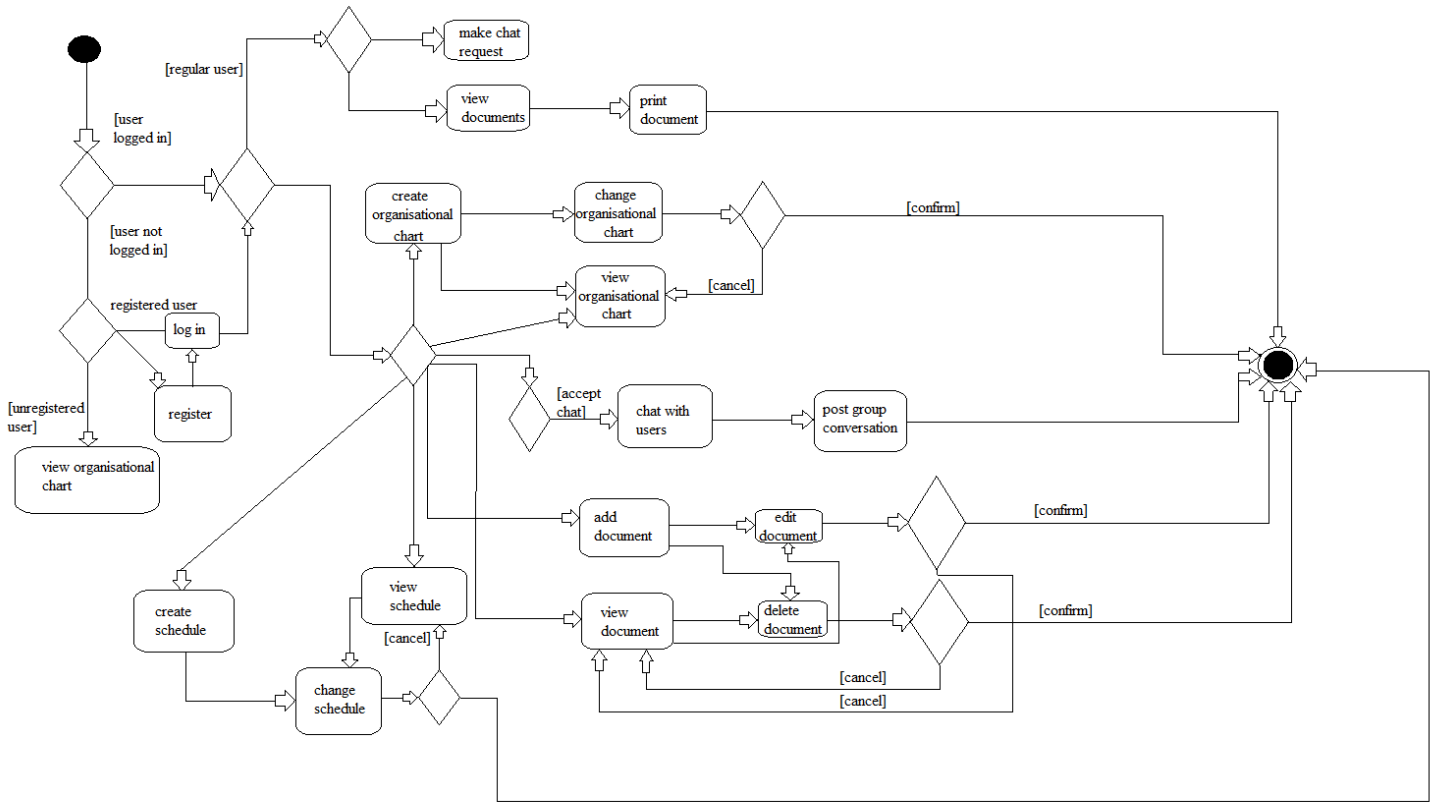
The product will have the following main functions:

- Presenting the organizational chart of the institutions of the municipality providing an institution level view and a department view.

- Serve as a platform for hosting meetings between a user/users and a municipality employee.
- Serve as online platform for document management.
- The municipality employees (administrators) can make changes to the organizational chart.
- The municipality employees(administrators) can set the online meeting hours.
- The municipality employees (administrators) can view edit and sign documents.
- The users can have an online audience with a municipality employee.



Use case diagram of the reviewing system.



Activity diagram

2.3 User Classes and Characteristics

The application users will be municipality employees, which will have administrator privileges, and citizens, which will have regular accounts.

2.4 Operating Environment

This application was developed in: Microsoft Visual Studio IDE.
The application supports the following web browsers Google Chrome and Mozilla Firefox.

2.5 Design and Implementation Constraints

This application is distributed with a proprietary software license.
The owner of proprietary software exercises certain exclusive rights over the software. The owner can restrict use, inspection of source code, modification of source code, and redistribution.

2.6 User Documentation

System Requirements Specifications document, System Design Specifications document and testing documents.

2.7 Assumptions and Dependencies

An internet connection is required in order to use the application.
Each specific data field must use a specific input data format.

3. External Interface Requirements

3.1 User Interfaces

The interface is meant to be simple, fast and user friendly.
The Layout should be the same for all users, with a main page for announcements and general information, and a navigation bar that allows access to the other functionalities and pages depending on the type of user.
The application general layout will contain:

The guest user interface will contain:

- View the organizational chart.
- View announcements

The registered user interface will contain:

- View the organizational chart.
- Establish an online (individual/group) meeting with a municipality employee.
- View the documents and announcements of the city hall.
- View announcements

The administrator interface will contain:

- Can modify audience schedule.
- Can modify documents.
- Can create/delete announcements.
- Respond to user message requests.

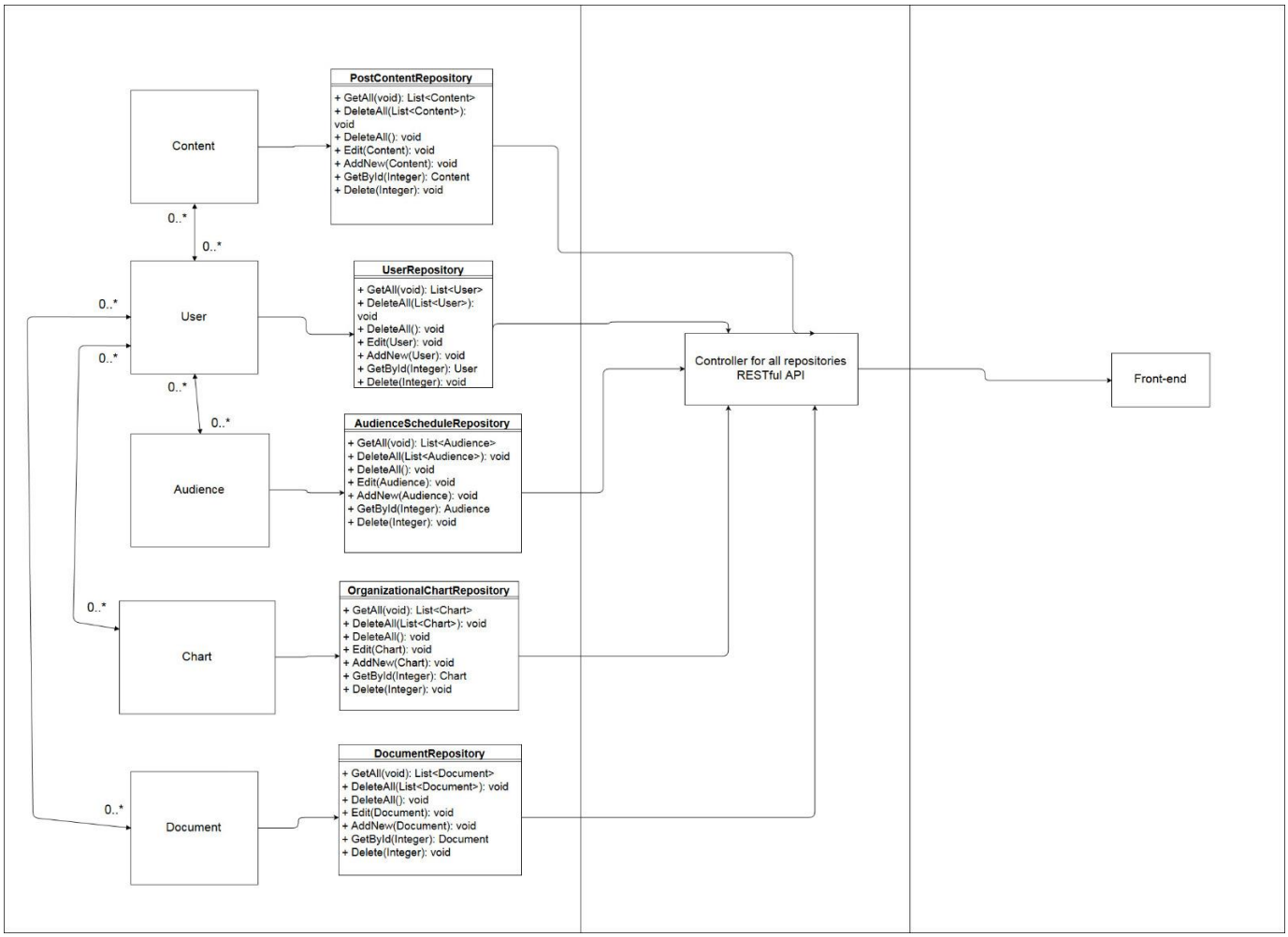
3.2 Hardware Interfaces

Not applicable.

3.3 Software Interfaces

There will be two functions that assure the communication between the server and the database: one to retrieve data from database (data RetrieveFromDatabase(int index, dbtable table)) and another one to insert data into the database(InsertIntoDatabase(Data data, dbtable table)).

There will be two functions that assure the communication between the client and the server: one to get data from the server (data RetrieveFromServer(int index)) and another one to send data to the server(SendToServer(Data data)).



3.4 Communications Interfaces

This application uses the HTTP protocol.

4. System Features

1. Authentication management:

1.1. Log in:

1.1.1. The log inpage shall enable a user to enter a user name and password.

1.1.2. The log in page must use the password field to keep the password from being viewable.

1.1.3. The log in page will indicate to the user if the user name and password were accepted.

1.1.4. After logging in the user should be redirected to the home page.

1.2. Register:

1.2.1. The register page shall allow a user to enter personal data in order to create a new account.

1.2.2. The register page must use the password field to keep the password from being viewable.

1.2.3. The register page will indicate if all fields were entered properly.

1.2.4. After registering the user should be redirected to the log in page.

1.3. Logout:

1.3.1. A logout process must be provided.

1.4. Password recovery:

1.4.1. The password recovery page shall enable the user to enter an email address.

1.4.2. Password recovery must validate the email address.

1.4.3. If the email address is validated successfully, a new password shall be sent.

2. Organizational chart management:

2.1. Town Hall departments will be chosen from a predefined name list.

2.2. New departments can be added.

2.3. Existing departments can be modified.

2.4. Existing departments can be removed.

2.5. Each department will have a graphical representation.

2.6. Relations between departments will be represented with orientated lines and annotated with the relationship type.

2.7. The graphical entity that will represent a department will allow adding information related to the people that will work in the department.

2.8. The graphical entity that will represent a department will allow adding information related to the relations between people.

2.9. The graphical entity that will represent a department will allow viewing information related to the people that will work in the department.

2.10. The graphical entity that will represent a department will allow viewing information related to the relations between people.

2.11. The organizational chart will be stored in a file with a predefined type.

3. Online audiences management:

3.1. Audience schedule will be specified for each week, with the possibility to be kept for the next week.

3.2. There can be more than two users (citizen and mayor) logged into the chat at the same time.

3.3. For group audiences the mayor will have the possibility to post the discussion on the website.

3.4. The audience planning option will allow users to make requests, in which they will specify their personal and contact data.

3.5. Following an audience request, users will receive an answer specifying the date and hour of the audience.

3.6. The application will automatically initiate a chat with the users at the planned date and hour, using the contact details provided.

4. Document management:

4.1. Files will be stored by categories.

4.2. The categories will be: informative documents, pending approval documents, modifiable pending approval documents.

4.3. There will be a function for searching documents in the database.

4.4. The search will be performed by document name, owner name or category.

4.5. The information system will allow scanning documents facilities.

4.6. The information system will allow printing documents facilities.

4.7. Each document will have a stamp of the last person who edited it.

5. Interfaces:

5.1. The clients will have an intuitive interface.

5.2. The clients will have a user-friendly interface.

5.3. The administrator will have an intuitive interface.

5.4. The administrator will have a user-friendly interface.

5.5. All icons will be created to have as little text as possible.

5.6. If a user entry must be one of several defined items, then the system shall present the user with the list of items from which to select the entry.

5.7. A maximum of 3 fonts shall be used in the user interface. These fonts shall be defined in the interface design.

5.8. A maximum of 4 font sizes shall be used in the user interface.

5.9. The pages will display the name of the logged on user.

6. Database:

6.1. The database server will be configured for storing data.

6.2. The database server will be configured for modifying data.

6.3. The database server will be configured for removing data.

6.4. Easy retrieval information from the database.

6.5. Possibility to modify the structure of the database.

6.6. Able to administer user rights.

6.7. Retrieve a report about which data has been modified, and indicating by whom.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

1. The information system will run on Windows and Unix OS.
2. Minimal system requirements:
   - Processor: 1 GHz
   - RAM Memory: 128 Mb
   - Storage space: 10 Mb
3. An internet connection should be set up before running the web application.

5.2 Safety Requirements

Not applicable.

5.3 Security Requirements

The sensitive information in this application is considered:

- User and password – password will be encrypted using hash;
- Conversations – using one way key, the key will be private;
- Database – Protection against SQL Injection will be set to every necessary input that works with the database.

5.4 Software Quality Attributes

Reliability: The system can be used by multiple users concurrently. Any user can access the system, even with a low performance PC.
Maintainability: The entire system shall be easy to maintain. System upgrades can quickly and safely be performed with a minimum of downtime.
Usability: The application will be user friendly, users can easily learn how to use it. The user interface can be accessed by differently abled users.

General programming rules:

Rule 1: Avoid using abbreviations. Exceptions: abbreviations commonly used as names, such as Id, Xml, Ftp.
Rule 2:Write only one statement per line.
Rule 3: No identifier in one name space should have the same spelling as an identifier in another name space, with the exception of structure member and union member names.
Rule 4: Add at least one blank line between method definitions and property definitions.
Rule 5: Add at least one blank line between method definitions and property definitions.
Rule 6: Place the comment on a separate line, not at the end of a line of code.
Rule 7: Use the + operator to concatenate short strings, as shown in the following code.

Rule 8: To avoid exceptions and increase performance by skipping unnecessary comparisons, use && instead of & and || instead of | when you perform comparisons, as shown in the following example.
Rule 9: Do not rely on the variable name to specify the type of the variable. It might not be correct.
Rule 10: Use implicit typing to determine the type of the loop variable in for and for each loops.

5.5 Business Rules

6. Other Requirements

Appendix A: Glossary

Appendix B: Analysis Models

The development of the application is based on the Agile strategy.

Appendix C: To Be Determined List

Not applicable.