

Software Requirements Specification for

“Distributed computer system for the management of an owner’s association”

**Prepared by :
Costache Theodor Alexandru
Dragoi Edith Cristiana
Enoiu Andra Maria**

March 10, 2021

Table of contents

1.	Introduction	4
1.1	Purpose.....	4
1.2	Product Scope	4
1.3	Intended Audience and Reading Suggestions.....	4
1.4	Definitions, acronyms, and abbreviations	4
1.5	References	4
2.	Overall Description	5
2.1	Product Perspective	5
2.2	Product Function.....	5
2.3	User Classes and Characteristics.....	6
2.4	Operating Environment.....	6
2.5	Design and Implementation Constraints.....	6
2.6	User Documentation	6
2.7	Assumptions and Dependencies	6
3.	External Interface Requirements.....	7
3.1	User Interfaces.....	7
3.2	Hardware Interfaces.....	8
3.3	Software Interfaces	8
3.4	Communications Interfaces	9
4.	System Features	9
4.1	Sign Up.....	9
4.1.1	Description Priority.....	9
4.1.2	Stimulus/Response Sequences	9
4.1.3	Functional Requirements.....	10
4.2	Sign In	10
4.2.1	Description Priority.....	10
4.2.2	Stimulus/Response Sequences	10
4.2.3	Functional Requirements.....	10
4.3	Log Out	11
4.3.1	Description Priority.....	11
4.3.2	Stimulus/Response Sequences	11
4.3.3	Functional Requirements.....	11
4.4	Sign In Admin	11
4.4.1	Description Priority.....	11
4.4.2	Stimulus/Response Sequences	11
4.4.3	Functional Requirements.....	11
4.5	Admin Add Admin	12

4.5.1	Description Priority.....	12
4.5.2	Stimulus/Response Sequences	12
4.5.3	Functional Requirements.....	12
4.6	Admin Delete User/Admin	12
4.6.1	Description Priority.....	12
4.6.2	Stimulus/Response Sequences	13
4.6.3	Functional Requirements.....	13
4.7	Admin Send Notification	13
4.7.1	Description Priority.....	13
4.7.2	Stimulus/Response Sequences	13
4.7.3	Functional Requirements.....	13
4.8	Admin Generate Reports.....	14
4.8.1	Description Priority.....	14
4.8.2	Stimulus/Response Sequences	14
4.8.3	Functional Requirements.....	14
5.	Other Nonfunctional Requirements	14
5.1	Performance Requirements	14
5.2	Safety Requirements.....	14
5.3	Security Requirements	15
5.4	Software Quality Attributes.....	15

1. Introduction

1.1 Purpose

This document has the purpose of giving a detailed description of the requirements of the project” Distributed Computer systems for the management of an owner’s association”.

1.2 Product Scope

This software will represent a web application representation of an owner’s association. The purpose is to simplify the activity of managing the tenants and to ease the owner’s access to information concerning their payments/invoices or anything related to the association.

1.3 Intended Audience and Reading Suggestions

This document is meant to be read by the developers and users.

1.4 Definitions, acronyms, and abbreviations

This Document was created based on the IEEE template for System Requirement Specification Documents:

- N/A – Not available
- Font – Times New Roman
- Dimension – 11
- The document is structured on chapters and each chapter in more subchapters.
- The language used for this document is English.

1.5 References

IEEE. IEEE Std 830-IEEE Recommended Practice for Software Requirements.

C# Net Application tutorial:

https://www.youtube.com/watch?v=GcFJpMFJvI&ab_channel=TraversyMedia

C# tutorial:

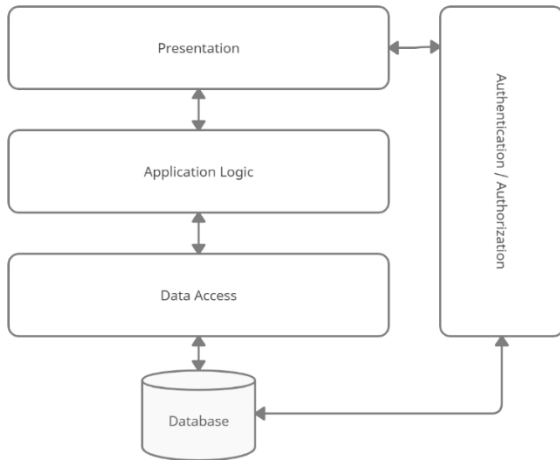
https://www.youtube.com/watch?v=GhQdIIFYlQ8&ab_channel=freeCodeCamp.org

Mysql database tutorial:

https://www.youtube.com/watch?v=7S_tz1z_5bA&ab_channel=ProgrammingwithMosh

2. Overall Description

2.1 Product Perspective



This product is meant as a standalone software, it is meant to be self-contained and non-referential to other products. A minimal requirement for the use of this product is access to the internet on the device used.

A database will be used to store the data and the application shall communicate with said database.

2.2 Product Function

These following functions are present in this product:

- **Common functionality:**
 - Login by means of Username and Password for Administrators/Owners.
 - User interface based on role.
 - Ability to see the payment debt for a specific apartment/outstanding debt from previous months.
 - Income/Expenses for the current month.
- **Administrator functionalities:**
 - Contract/Employee editing/addition.
 - Report of sums to be collected from the owners.
- **Customer functionalities:**
 - Addition of bills for the current month (on top of the ones that already exist).
 - Addition of water consumption per apartment.
 - Actualization of occupied/unoccupied apartments and the number of tenants in each apartment.
 - Addition of bills such as the rent collected from tenants.

2.3 User Classes and Characteristics

Administrator class which will have the role of administrating the system. They have the ability of managing the entire application. Tenants which shall take the role of customer, having the ability to see invoices, information on their monthly incomes/expenditures.

2.4 Operating Environment

Any person who will use this application must have the following:

- Microsoft Windows XP at least installed
- One of the next modern browsers :
 - Google Chrome (v 82.0)
 - Mozilla FireFox (v 73.0)
 - Microsoft Internet Explorer 11,10,9
 - Microsoft Edge (v 80.0.361.62)

2.5 Design and Implementation Constraints

Constraints:

- Computers will have to be connected to the same SQL server 2017 or newer.
- The developers shall use the following apps: Visual Studio 2019, Entity Framework Core, Visual Code.

2.6 User Documentation

This word document represents the documentation for the project.

2.7 Assumptions and Dependencies

Assumptions:

- All computers that wish to connect shall use the same SQL Server 2017 or newer.
- Access to internet and a web browser (Google Chrome v 82.0; Mozilla FireFox v 73.0; Microsoft Internet Explorer 11/10/9; Microsoft Edge v 80.0.361.62).

3. External Interface Requirements

3.1 User Interfaces

There shall be 2 interfaces, one for the administrator and one for the user. A common interface, the login screen, which will have an input prompt for the username and password, which will be present for both roles.

The layout shall be as such:

The HEADER containing:

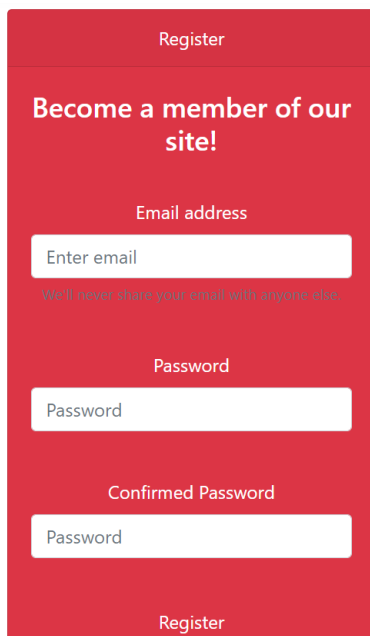
- The logo
- The currently logged in user's username.
- Buttons for log-out and account information ("My account").
- Button for the checking of invoices ("Invoices").
- Button for checking info pertaining to income/debts/expenditure (Services).

The body shall be different pertaining to the Invoices and Services buttons which shall take the user to a different UI depending on their role.

The body shall contain:

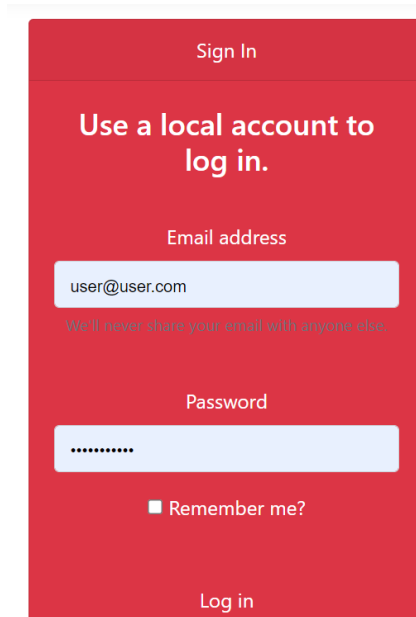
- The menu.
- The content.

SignUp Menu:



The Register form is a red rectangular card. At the top, it has a title bar with the word "Register" in white. Below this, the main heading "Become a member of our site!" is centered in white. The form contains three input fields: "Email address" with a placeholder "Enter email", "Password" with a placeholder "Password", and "Confirmed Password" with a placeholder "Password". Each input field has a light blue border and a small red error message "We'll never share your email with anyone else" below it. At the bottom, there is a "Register" button in white.

SignIn Menu:



The Sign In form is a red rectangular card. At the top, it has a title bar with the words "Sign In" in white. Below this, the main heading "Use a local account to log in." is centered in white. The form contains two input fields: "Email address" with a placeholder "user@user.com" and "Password" with a placeholder ".....". Each input field has a light blue border and a small red error message "We'll never share your email with anyone else" below it. Below the password field, there is a checkbox labeled "Remember me?". At the bottom, there is a "Log in" button in white.

3.2 Hardware Interfaces

N/A

3.3 Software Interfaces


The application shall be connected to a database (in this case an SQL Server) which will contain all the necessary information.

TennantsAssociation

Home


Register

Login



Welcome to tenants association

To see your invoice you have to be logged in. Please click on the button to sing in. If you don't have an account please create one.



TennantsAssociation

Home

Manage Invoices

Reports

Profile

Logout

Invoices

Create New

Id Invoice	User Name	Apartment Number	Price	Status	
5	user@user.com	1	123	Paid	<a>Edit <a>Details <a>Delete
6	user@user.com	1	345	Unpaid	<a>Edit <a>Details <a>Delete
7	user@user.com	1	140	Unpaid	<a>Edit <a>Details <a>Delete
8	user1@user.com	2	200	Unpaid	<a>Edit <a>Details <a>Delete

TennantsAssociation

Home

Pay Invoices

Building

Apartment

Profile

Logout

Building

Add new building details

StreetName	StreetNo	BuildingNo	
street	1	1	<div>EditDetailsDelete</div>
street2	2	2	<div>EditDetailsDelete</div>

TennantsAssociation

Home

Pay Invoices

Building

Apartment

Profile

Logout

Apartments

Create New

ApartmentNo	UserName	BuildingNo	Building	
1	user@user.com	1	1	<div>EditDetailsDelete</div>
2	user@user.com	2	2	<div>EditDetailsDelete</div>
1	user@user.com	2	1	<div>EditDetailsDelete</div>
1	user@user.com	2	1	<div>EditDetailsDelete</div>

3.4 Communications Interfaces

HTTP will be the communication protocol that our app will use.

4. System Features

4.1 Sign Up

4.1.1 Description Priority

This feature will be used by the user to create an account. Without an account the user will not be able to access the application and its other features.

4.1.2 Stimulus/Response Sequences

The user has to input a username, an email address, a password and retype the password for confirmation. After all this informations are required by the user and press the sign up button then the informations are stored in the database and the account is created.

4.1.3 Functional Requirements

- 4.1.3.1 The sign up will be used by the users to register.
- 4.1.3.2 Username text box: The user can input his username as text.
- 4.1.3.3 If the username already exists, then a message will be displayed.
- 4.1.3.4 Password text box: The user can input his password as text, numbers or symbols.
- 4.1.3.5 The password will be encrypted in the database for protection.
- 4.1.3.6 Confirmation password text box: The user can retype his password to confirm it.
- 4.1.3.7 If the confirmed password is not the same with the password above, then an error message will be displayed.
- 4.1.3.8 Email text box: The user can input his email to create the account.
- 4.1.3.9 If the email is not correct, then an error message will be displayed.
- 4.1.3.10 Sign up button: Using this button, after the user introduced all the information required (username, email and password), the account will be created.

4.2 Sign In

4.2.1 Description Priority

This feature will be used by the user to log in to the account. Without logging in the user will not be able to access the application and its other features. To use this feature the user has to be registered.

4.2.2 Stimulus/Response Sequences

The user has to input the email and the password for the account. After the informations are required by the user and press the sign in button then the database is accessed and the email and the password will be searched. Once the user is found in the database then the user can access his account.

4.2.3 Functional Requirements

- 4.2.3.1 The sign in will be used by the users to log in to the account.
- 4.2.3.2 Email text box: The user can input their email to be checked.
- 4.2.3.3 If the email is not correct, then an error message will be displayed.
- 4.2.3.4 Password text box: The user can input their password to be checked.
- 4.2.3.5 If the password is not correct, then an error message will be displayed.
- 4.2.3.6 Sign in button: Using this button, after the user introduced all the information required (email and password), the user can log in to the account.

4.3 Log Out

4.3.1 Description Priority

This feature will be used by the user to exit his account. To use this feature the user has to be registered and logged in to the account.

4.3.2 Stimulus/Response Sequences

The user can log out from his account menu using the log out button, if the user is sign in to the account. The log out feature will not delete the information from the data base.

4.3.3 Functional Requirements

- 4.3.3.1 The user can use this feature only if it has an account and is signed in.
- 4.3.3.2 Once the user is signed in the features for sign up and sign in will not be available anymore.
- 4.3.3.3 The informations from databases will not be deleted.

4.4 Sign In Admin

4.4.1 Description Priority

This feature will be available only for admin. The admin will be already registered in the database and in the database it will be an attribute that indicates whether it is admin or not.

4.4.2 Stimulus/Response Sequences

The admin will input his email and password and by the information stored in the database and the specific attribute for the admin, he/she will be able to access the admin account.

4.4.3 Functional Requirements

- 4.4.3.1 This type of sign in will be used only by the admins.
- 4.4.3.2 Email text box: The admin can input their email to be checked.
- 4.4.3.3 If the email is not correct, then an error message will be displayed.
- 4.4.3.4 Password text box: The admin can input their password to be checked.
- 4.4.3.5 If the password is not correct, then an error message will be displayed.
- 4.4.3.6 Sign in button: Using this button, after the admin introduced all the information required (email and password), the admin can log in to the account.
- 4.4.3.7 When sign in it will be looked after the specific attribute for the admin in the database, which makes the difference between the user and the admin.

4.5 Admin Add Admin

4.5.1 Description Priority

This feature will be used only by the admins. Here one admin can add another admin.

4.5.2 Stimulus/Response Sequences

The admin will provide the necessary informations to add another admin: email, password and username. The informations will be stored in the database. The new admin will be able to change the password.

4.5.3 Functional Requirements

- 4.5.3.1 The admins will be using this to add other admins.
- 4.5.3.2 Username text box: The admin can input the username as text.
- 4.5.3.3 If the username already exists, then a message will be displayed.
- 4.5.3.4 Email text box: The admin can input the email as text.
- 4.5.3.5 If the email is not correct, then an error message will be displayed.
- 4.5.3.6 Password text box: The admin can input the password as text, numbers or symbols.
- 4.5.3.7 If the password is not correct, then an error message will be displayed.
- 4.5.3.8 The new admin will be able to change the password.

4.6 Admin Delete User/Admin

4.6.1 Description Priority

This feature is used by one admin to delete an other admin or an user.

4.6.2 Stimulus/Response Sequences

The admin has the list with users and the list with admins from the application. For each one, the admin, has a set of actions that can be done for them (including deleting an user/admin). The Admin Delete User/Admin feature will delete the information from the data base.

4.6.3 Functional Requirements

4.6.3.1 The admin can delete an user or an admin.

4.6.3.2 The admin will have a list of users/admins.

4.6.3.3 In each list the admin will have a button (“Delete”) which can be used to delete the user/admin.

4.6.3.4 The system shall inform the user or the admin when any of the changes above are made.

4.6.3.5 The informations from database will be deleted.

4.7 Admin Send Notification

4.7.1 Description Priority

This feature will be used by the admin to send to an user notification like invoices, unpaid bills, over due date.

4.7.2 Stimulus/Response Sequences

The admin will complete a form with necessary informations as the sum of the invoice, the due date, the apartment, the name of the user. Once the form is completed and sent to the user, the user will receive a notification with this informations.

4.7.3 Functional Requirements

4.7.3.1 Ability to allow a logged in admin to add/modify an invoice.

4.7.3.2 Input tab for the user's username.

4.7.3.3 Input tab for the text of the invoice.

4.7.3.4 Input tab for the sum of the invoice.

4.7.3.5 Input tab for the due date of the invoice.

4.7.3.6 The ability to send an invoice to a specific user.

4.7.3.7 The user shall be notified of the incoming invoice.

4.8 Admin Generate Reports

4.8.1 Description Priority

This feature will be used by the admin to generate reports about the invoices, due dates or unpaid bills.

4.8.2 Stimulus/Response Sequences

The admin can see what user pay or didn't pay the bill, if there are any over due dates.

4.8.3 Functional Requirements

4.8.3.1 In the menu there will be a "Reports" menu with Income, Overdue and Paid reports.

4.8.3.2 Every report will show the name of the user and the sum of the bill, as well as the due date.

4.8.3.3 Every report will say if the bill has been paid or not.

4.8.3.4 If the bill is overdue and is not paid then the user will receive a notification.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

The product shall take initial load time depending on the internet connection strength which also depends on the media-device from which the product is run. The performance shall depend upon hardware components of the client/customer.

5.2 Safety Requirements

N/A

5.3 Security Requirements

When the user/admin enters the password, it shall not be visible. The password shall be encrypted in the database.

5.4 Software Quality Attributes

Following rules shall be respected throughout the implementation code:

- Source code shall be commented appropriately to minimize learning effort for further development plans.
- Only one declaration/function call shall be used on a single line.
- Variables and Functions shall have non-abstract names, only meaningful names shall be used.
- Functions should be broken down as much as possible. A function should have an appropriate number of operation calls.
- Unreachable code can exist within certain metrics.