

University of Latvia

Faculty of Computing

Course code and title: DatZ2072 “Software Engineering”

Social Memories

Scientific advisor: Mg.sc.comp. J.Iljinš

Done by:

Raisa Nazarova (rb05009),

Alina Kalinina (ak14113),

Nikita Kuznecovs (nk21002),

Artem Osipov (ao21001),

Karlis Akots Gribulis (kg16029),

Riga, 2021

Abstract (ENG)

Project work describes “Social memories”, a platform which is based on a regular diary basis, where you can save notes, thoughts and share them directly to the platform, or you can connect it with famous platforms such as Twitter, Facebook, Instagram and others. You can hide your notes just for yourself, share with your friends via personal link or make it just visible for everyone. Guests can just view public diaries, registered users can see, like, comment posts in other diaries as well as send personal diary links to their friends and administrators can approve (green tick near the post) or delete inappropriate posts as well as ban users who violate platform rules several times. System consists of 3 parts: registered users, unregistered users or guests and administrators. The document is developed according to the ISO/IEC/IEEE 29148:2018 standards.

Keywords: notes, memories, social, thoughts, diary.

Anotācija (LV)

Projekta darbs apraksta “Social memories”, tā ir platforma, kura ir balstīta uz dienasgrāmatas pamata, kur lietotājs var saglabāt piezīmes, domas un dalīties ar tām platformā, vai arī jūs varat ar tām dalīties citās populārās platformās, kā, piemēram, Twitter, Facebook, Instagram un citas. Jūs varat savas piezīmes paturēt privātas vai dalīties ar saviem draugiem, kā arī izveidot piezīmes publiski pieejamas visiem lietotājiem. Viesi var redzēt publiskās dienasgrāmtas, savukārt, reģistrētie lietotāji var redzēt, komentēt, atzīmēt “patīk” citu dienasgrāmatu ziņas, kā arī sūtīt savas privātās dienasgrāmatas tiešsaistes adresi saviem draugiem un administratori var apstiprināt (zaļa atzīme blakus rakstam) vai arī izdzēst nepiemērotus rakstus, kā arī aizliegt lietotāju pieeju platformā ja viņi ir vairākkārtīgi pārkāpuši platformas noteikumus. Sistēma sastāv no 3 daļām: reģistrētie lietotāji, neregistrētie lietotāji, administratori. Dokuments ir izstrādāts saskaņā ar ISO/IEEE 29148:2018 standartu.

Atslēgas vārdi: piezīmes, atmiņas, sociāls, domas, dienasgrāmata.

Table of Content

Introduction	5
1. Overall description	7
1.1. Current situation	7
1.2. Client	7
1.3. Product vision	7
1.4. Business requirements	8
1.5. System Users	8
1.6. General restrictions	9
1.7. Assumptions and dependencies	10
2. Software requirements specification	11
2.1. Conceptual Database Model	11
2.2. Functional Requirements	12
2.2.1. General description of functions	12
2.2.2. Functional breakdown by modules	13
2.2.3. User Management module	15
2.2.4. Admin management module	23
2.2.5. Guest management module	32
2.2.6. Data Submission module	35
2.2.7. Notifications module	42
2.3. Non-functional requirements	51
3. Software design description	52
3.1. Database design	52
3.1.1. User and corresponding entities	54
3.1.2. Admins and corresponding entities	56
3.1.3. Guests and corresponding entities	57
3.1.4. Data and corresponding entities	57
3.1.5. Notifications and corresponding entities	58

3.2. Partial Design of User Interfaces	60
3.3. Partial Functional Design	71
References and sources	76

Introduction

Purpose

The document describes “Social memories” web application general vision of the application, its users, software requirements specifications, functionality and design of the system. This document is a mutual agreement between the customer and the contractor.

Scope

“Social Memories” web app expands the meaning of a personal diary. While staying public it has enough privacy for those users who are especially seeking it. Plus, it is free and does not require any financial spendings. The app is simple, intuitively understandable, user-friendly but still has enough functionality for those who are looking for a better way of storing their own thoughts, saving memories or just constructing the list of favourite movies.

Links with other documents

The document is created according to the ISO/IEC/IEEE 29148:2018 standart.

As it is a public platform for different types of users with unique backgrounds, views, ages, etc. then the Internet rules may apply :)

Overview

The document consists of 3 chapters, the description of the system, the requirements, definitions, acronyms, abbreviations, references and sources and appendices.

- Chapter 1 “Overall description” telling about current situations with demand about new application, showing clients who will use this system, product vision of the developers, business requirements, explaining system users roles also explaining general restrictions , assumptions and dependencies
- Chapter 2 “Software requirements specifications” telling about conceptual database model, functional and non-functional requirements. In this section are described such

modules as: User Management module, Apartment Management module, Inhabitants Management module, Data Submission module, Billing and Notifications modules.

- Chapter 3 “Software design description” describes database designs.
- Appendix contains information excluded from the main sections due to being additional.

1. Overall description

It is hard to find a person who has never wanted to write something down. The recipe, the name of the song heard over the radio, a funny joke or just personal emotions, thoughts and ideas. In many situations you don't have any paper to write it down but phone planners or notes allow only offline storage for such things.

"Social Memories" allows users not only to save their needed information both online and offline but also make it public-private at any time they decide and as well share it at once in every popular social media platform like Facebook, Twitter, Instagram and so on.

1.1. Current situation

There are several similar applications on the market but it is social media applications. The biggest downside of all of them is that they are social media and users can use it only online. Users can share their posts with friends but only while they are online.

Application "Social memories" for now will be the only one. Users will be able to record their thought offline and online. It is not social media it is social notes for users thoughts. If users want, they can share their thoughts to another social media, but if users want to stay alone with themselves they can be offline and write only their thoughts, saving them.

1.2. Client

The system is developed for any Internet user but the target audience for the diary is between 17 and 30 years old individuals. Young people are more likely to create a diary and put their own thoughts, feelings and ideas inside.

1.3. Product vision

The main objective of the project is to give opportunities on a regular diary basis, where you can save notes, thoughts and share them directly to the platform, establish a

network among the people residing in a specific community and all the information can be easily accessed and shared among the people.

This system allows users to register their various types of profile like social thanks to all available social network platforms and personal.

This system allows users to maintain the scrap book whatever scrap he has sent to users.

This system allows users to join the communities according to their scenario.

This system provides the user to maintain their friend list and users can update their friend list.

This system allows users to send invitations to another friend to join the community or group and can add to their friend list for future.

1.4. Business requirements

The application provides stable, available 24 hours and user friendly UI service to express users thoughts privately or sharing the post, which is making the company more attractive for new users and services.

1.5. System Users

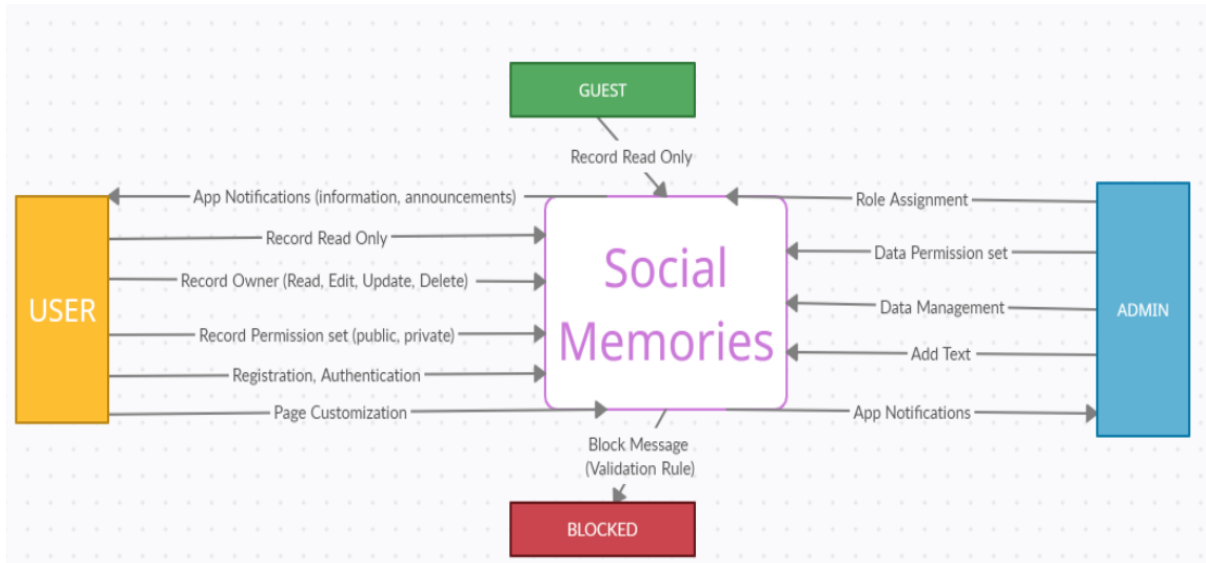
There are 4 system user roles, and a system user can only be a part of one of the following roles:

- **Guest** - most basic user role that can only view public content of the website and it cannot comment, like any posts, but this user has the ability to register and be granted a **User** role.
- **User** - a registered person that has the same abilities as a **Guest** and also can view private posts (if shared to him) and there are no comment or liking restrictions (unless limited by an **Administrator**). On top of that, **User** can create his/her own public or private posts.
- **Administrator** - most privileged role and is only granted to the platform administrators that are responsible for checking if no rules are broken in other people's posts, comments, usernames, etc. **Administrator** has the ability to

approve posts becoming public or remove posts and comments or even block a user, if the user has violated platform rules.

- **Blocked** user - most unprivileged user role, this user cannot interact with the platform posts or platform itself in any form. **User** can become a **Blocked** user if he/she has violated platform rules and an **Administrator** has blocked him/her.

All data flow between user and application are shown in the diagram of 0 level (pic.1.).



Pic. 1. Data flow diagram

1.6. General restrictions

As the system is a web application it's mandatory for end-users to have a device with connection to the internet with installed web browser. System providers also must have a delegated internet domain as well as server for storing the application data.

Post restrictions are as follows: no 18+ topics, for example, gore, nudity, sexual content, no racism, harassment, extremism, bullying, racial discrimination or other offensive attitude towards any owner of the diary or other users. There are other restrictions, such as, the need to have appropriate usernames (no slurs, sexual, racist or similar connotations).

New Users are checked by the administrators after creation and username cannot be changed. Administrator has to approve topics before they can become public (green tick near the theme name).

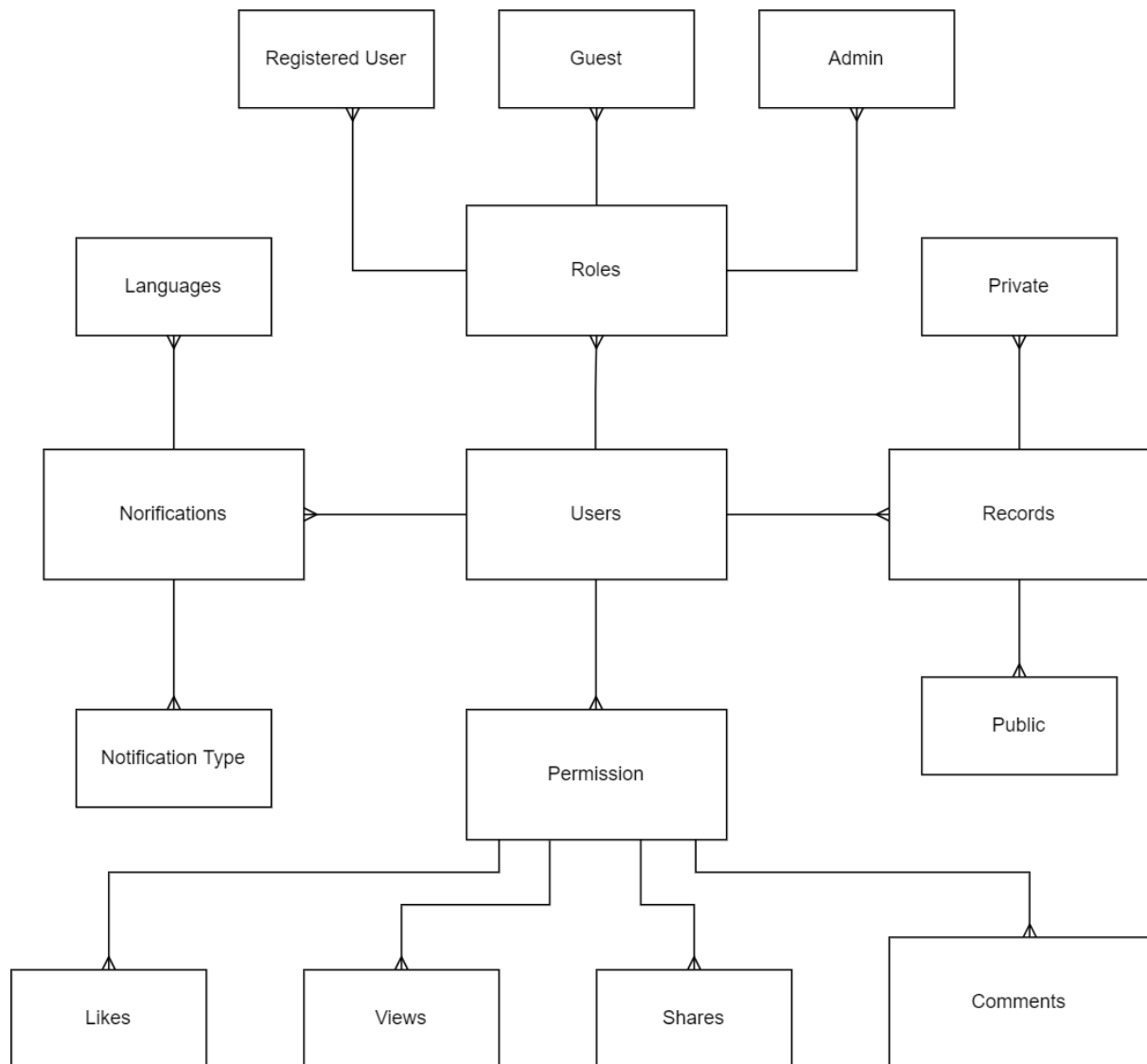
1.7. Assumptions and dependencies

It is assumed that during the setup of the system there will be a database in place with already created base tables and information, such as, user types, permissions, post visibility, etc. so the system could be properly used from the start. Also, it is expected that there will already be a pre-selected team of administrators that are well informed on how the system works and all its guidelines. For proper and high quality service the system depends on a support team which will address any issues or bugs in it.

2. Software requirements specification

2.1. Conceptual Database Model

A Conceptual Data Model is an organized view of database concepts and their relationships. The purpose of creating a conceptual data model is to establish entities, their attributes, and relationships (pic. 2.).



Pic. 2.. Conceptual database model

- Entity “Roles” is an entity for “Users” and contains the list of user roles and descriptions.
- Entity “Guests” has information about non registered users.
- Entity “Users” has information about user nickname, password, email, user role and email notifications status.

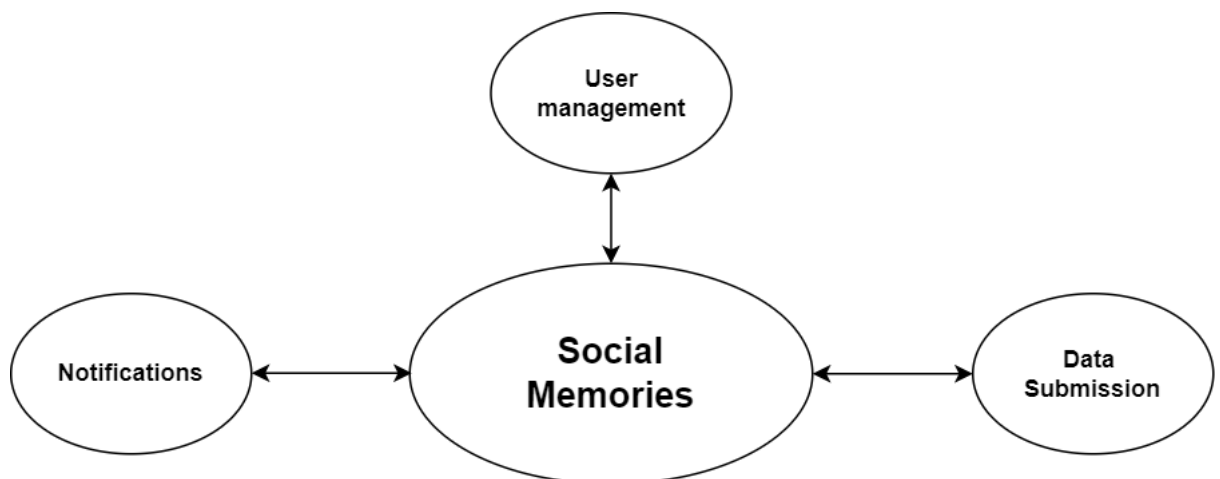
- Entity “Notifications” has information about the users and messages.
- Entity “Notifications type” containing a list of possible notification types.
- Entity “Languages” containing a list of possible languages.
- Entity “Public records” containing information about public records from users.
- Entity “Private records” containing information about private records from users.
- Entity “Permission set” containing information about all permissions.
- Entity “Comments” containing information about comments from users.
- Entity “Banned” containing information about banned users.
- Entity “Blocked” containing information about blocked users.

2.2. Functional Requirements

2.2.1. General description of functions

The system consists of 3 modules: User management, data submission and notification. High-level general description conceptual model structure is shown on the *pic. 3*.

Description of modules functions and corresponding user groups with access permissions is described in *table 1*.



Pic. 3 Modules of the system

Table 1. Modules functions and corresponding user groups

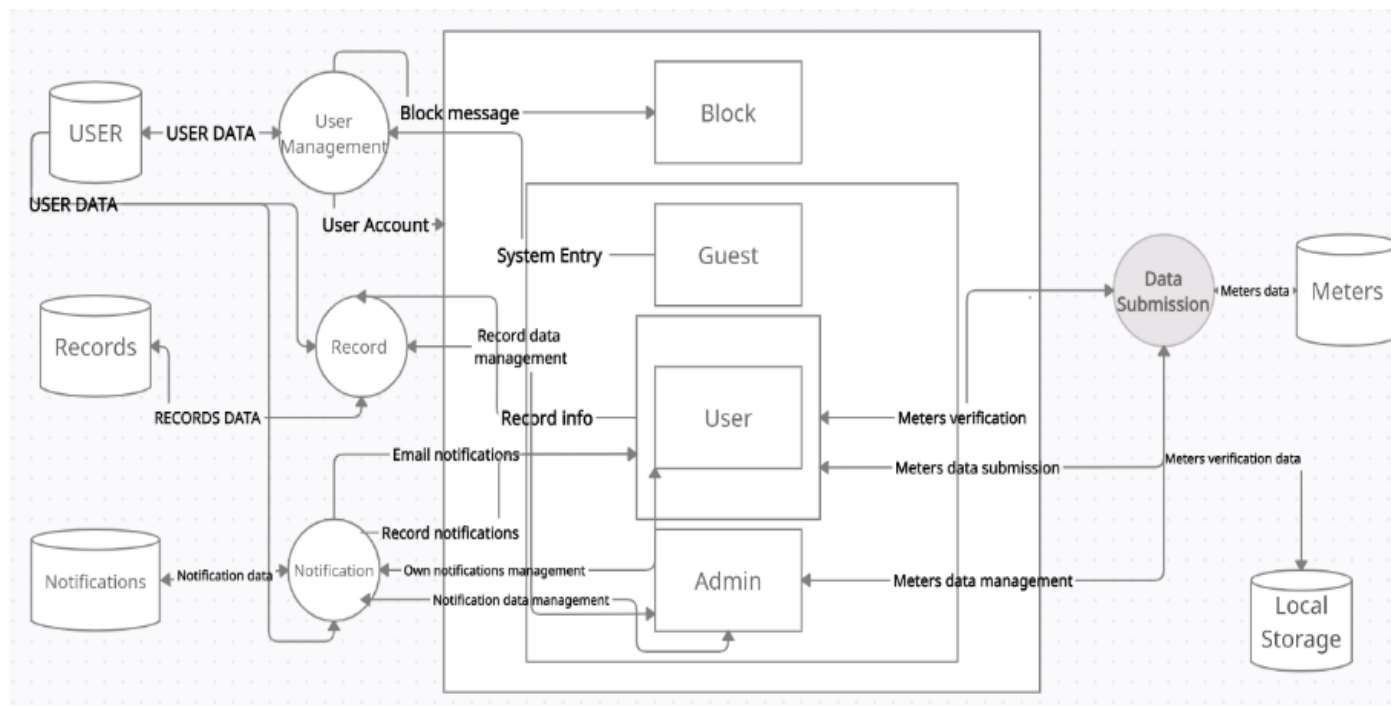
Module	Function	User group
User module	Registration	Guest
	Authentication	Guest
	Page customization	User, Admin
	Permissions	Admin
	View user list	Admin
	User role changing	Admin
	Logout	User, Admin, Blocked
Notification module	Public view	Guest, User, Admin
	Private view	User, Admin
	Create note	User, Admin
	Delete note	Admin
	Delete private note	Admin, User
	Edit private note	Admin, User
	Edit public note	Admin
	Receive notifications about new note	Admin, User
	Disable/enable notifications	User
Data submission module	View data information	User
	View all data information	Admin
	Upload data	User
	Edit data	User
	Delete data	Admin, User
	Create data	User
	Update data	User

2.2.2. Functional breakdown by modules

The data flow diagram (*pic.2.2.2.*) displays functional breakdown by modules based on the description on the table 2.2.1. and module in section 2.2.1.

Several modules are interconnected with each other and require data from another module to be able to proceed. These modules are:

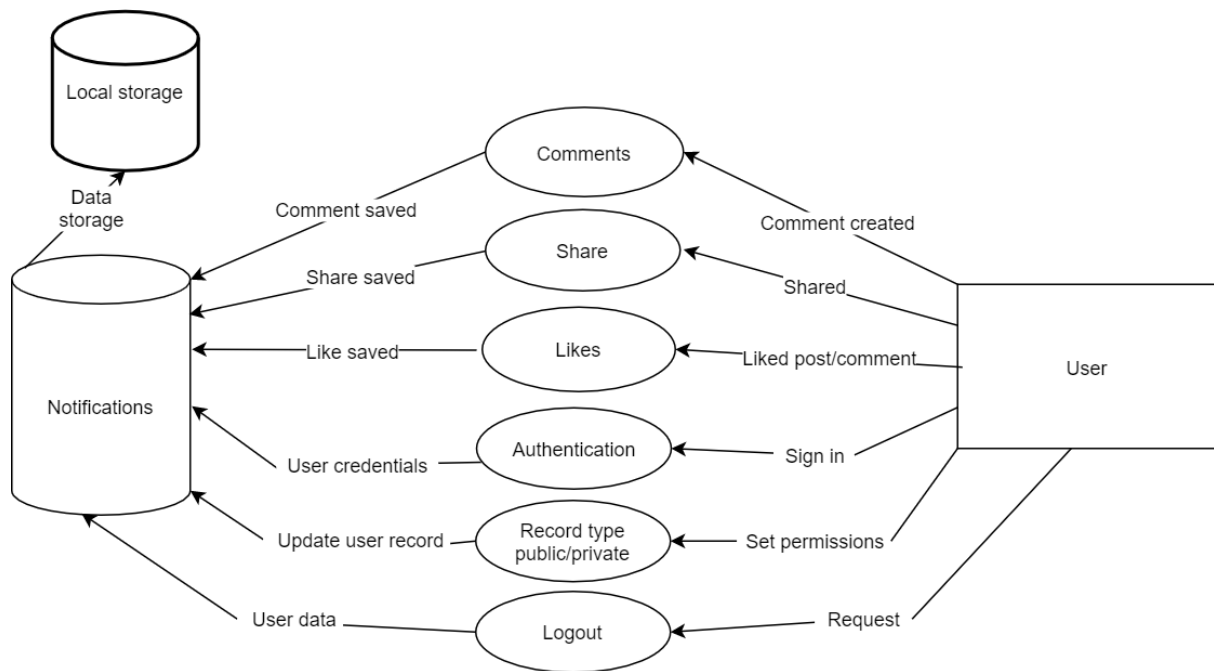
- User management module requires data about users and users roles.
- Guest management module requires data about non registered users.
- Admin management module requires data about records, notifications, users information, system.
- Data management module requires data about records, users, notifications, records.
- Notifications module requires data from data module to send notifications to users.



Pic.4. Data flow diagram

2.2.3. User Management module

This section goes over each function described in *table 2* and section 2.2.1. Each function is described separately in *tables 2-8* respectively.



Pic. 5. User management module - data flow diagram (level 2)

Table 2. User Management module - registration function.

ID	User group
REGISTRATION	Guest
Name	
Registration	
Description	
New user registration by a guest	

Pre-conditions
<ol style="list-style-type: none"> 1) Not logged in the system 2) Account does not exist
Input
<ul style="list-style-type: none"> • User's name - up to 255 characters • User's email - up to 255 characters • User's password - up to 255 characters, minimum 8 characters
Processing
<ol style="list-style-type: none"> 1) Check if email contains "@" in it <ol style="list-style-type: none"> a) if not, return error message (1) 2) Check if user's name is less than 255 characters <ol style="list-style-type: none"> a) if not, return error message (4) 3) Check if email is not longer than 255 characters <ol style="list-style-type: none"> a) if not, return error message (5) 4) Check if email is already registered <ol style="list-style-type: none"> a) if exists, return error message (6) 5) Check password length <ol style="list-style-type: none"> a) if less than 8 characters, return error message (2) b) if longer than 255 characters, return error message (3) 6) Check password confirmation field <ol style="list-style-type: none"> a) if is not same as password field, return error message (7) 7) Add new user to database
Post-conditions
<ol style="list-style-type: none"> 1) Registered in the system 2) Logged in the system
Outputs
If user is successfully added to the system, he will be automatically logged in the system
Error messages
<ol style="list-style-type: none"> 1) Please include '@' in the email address.

- 2) The password must be at least 8 characters.
- 3) The password may not be greater than 255 characters.
- 4) The name may not be greater than 255 characters.
- 5) The email may not be greater than 255 characters.
- 6) The email has already been taken.
- 7) The password confirmation does not match.

Table 3. User Management module - authentication function.

ID	User group
AUTHENTICATION	Guest
Name	
Authentication	
Description	
Logging in to the system by a guest	
Pre-conditions	
1) Not logged in the system	
Input	
<ul style="list-style-type: none"> • User's email - up to 255 characters • User's password - up to 255 characters, minimum 8 characters • Remember me (optional) 	
Processing	
<ol style="list-style-type: none"> 1) Check if email contains "@" in it <ol style="list-style-type: none"> a) if not, return error message (1) 2) Check if email is not longer than 255 characters <ol style="list-style-type: none"> a) if not, return error message (4) 3) Check password length <ol style="list-style-type: none"> a) if less than 8 characters, return error message (2) b) if longer than 255 characters, return error message (3) 	

4) Check if email and password matches record a) if does not, return error message(5)
Post-conditions
1) Logged in the system 2) If user's role is Blocked - show error message
Outputs
If the user is successfully logged in, he sees his personal post page.
Error messages
1) Please include '@' in the email address. 2) The password must be at least 8 characters. 3) The password may not be greater than 255 characters. 4) The email may not be greater than 255 characters. 5) These credentials do not match our records.

Table 4. User Management module - page customization function.

ID	User group
PAGE CUST	User, Admin
Name	
Page customization	
Description	
Customizing page layout.	
Pre-conditions	
1) Logged in the system 2) User does not have Blocked role	

3) Opened page settings
Input
<ul style="list-style-type: none"> • Profile picture (optional) • Posts per page (optional) • Font size (optional) • Background color (optional) • Do not show post author (optional) • Sound on new comments (optional) • Sound on new notification (optional)
Processing
1) Changes user page layout preferences in database
Post-conditions
1) Page gets refreshed and has its view updated
Outputs
Page with updated view based on selected preferences.

Table 5. User Management module - permission function.

ID	User group
USER PERMISSIONS	Admin
Name	
Permissions	
Description	
Changes permissions of the individually selected User	
Pre-conditions	

1) Logged in to the system as Admin 2) Opened list of all users by View user list function (<i>table 6</i>) 3) Select function “Change permission” for selected user
Input
<ul style="list-style-type: none"> • User read permissions • User post permissions • User comment permissions • User delete post permissions
Processing
1) Update record field “user permissions” for the selected user in database, based on the selected permissions, chosen from drop-down list
Post-conditions
1) User permissions get updated in database
Outputs
Returns a formatted list of all users of the system (by View user list function, table 6) with changed user permissions.

Table 6. User Management module - view user list function.

ID	User group
USER LIST	Admin
Name	
View user list	
Description	
Proceed and shows the list of all users of the system with corresponding user roles and permissions	

Pre-conditions
1) Logged in to the system as Admin 2) Clicked on “View users” button
Processing
1) Selects all Users from database 2) Adds “Change role” (<i>table 7</i>) and “Change permissions” (<i>table 5</i>) function near each user
Post-conditions
1) Full user list and function shown
Outputs
Returns a formatted list of all users of the system, consisting of user’s name, email, role, permission and “Change role” function (<i>table 7</i>), and “Change permissions” (<i>table 5</i>). There always will be at least 1 user – admin itself.

Table 7. User Management module - user role changing function.

ID	User group
USER ROLES	Admin
Name	
User role changing	
Description	
Changes the role of the selected user to different to add pre-set permissions.	
Pre-conditions	
1) Logged in to the system as Admin 2) Opened list of all users by View user list function (<i>table 6</i>)	

3) Select function “Change role” for selected user
Input
<ul style="list-style-type: none"> • User role
Processing
1) Updates record field “user role” for the selected user in the database, based on the role chosen from the drop-down list.
Post-conditions
1) User role and permissions get updated in database
Outputs
Returns a formatted list of all users of the system (by View user list function, table 6) with changed user role.

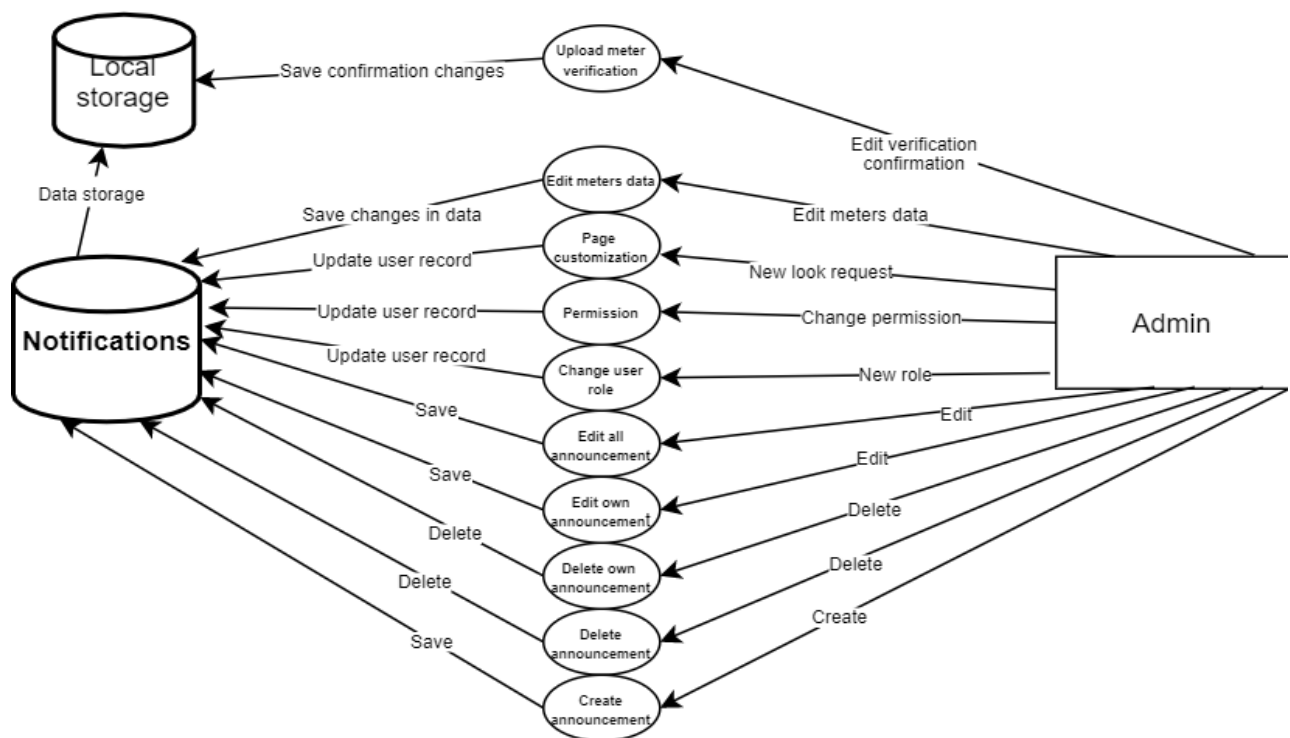
Table 8. User Management module - logout function.

ID	User group
LOGOUT	User, Admin, Blocked
Name	
Logout	
Description	
Logout from the system by any non-guest user	
Pre-conditions	
1) Not be a Guest 2) Logged in to the system	

Input
<ul style="list-style-type: none"> Click “Logout” button
Processing
1. Log out from the system with session manipulation
Post-conditions
1) Logged out of the system
Outputs
Main page of the system.

2.2.4. Admin management module

The data flow diagram of the Admin management module (pic. 6) is based on the function described in *table 1*.



Pic.6. Admin management module - data flow diagram (level 2)

Table 9. Admin Management module - registration function.

ID	User group
REGISTRATION	Guest
Name	
Registration	
Description	
Admin registration by a guest	
Pre-conditions	
3) Not logged in the system 4) Account does not exist	
Input	
<ul style="list-style-type: none"> • User's name - up to 255 characters • User's email - up to 255 characters • User's password - up to 255 characters, minimum 8 characters 	
Processing	
8) Check if email contains "@" in it b) if not, return error message (1) 9) Check if user's name is less than 255 characters a) if not, return error message (4) 10) Check if email is not longer than 255 characters a) if not, return error message (5) 11) Check if email is already registered a) if exists, return error message (6) 12) Check password length a) if less than 8 characters, return error message (2) b) if longer than 255 characters, return error message (3) 13) Check password confirmation field a) if is not same as password field, return error message (7)	

14) Add new user to database
Post-conditions
3) Registered in the system 4) Logged in the system
Outputs
If admin is successfully added to the system, he will be automatically logged in the system
Error messages
8) Please include '@' in the email address. 9) The password must be at least 8 characters. 10) The password may not be greater than 255 characters. 11) The name may not be greater than 255 characters. 12) The email may not be greater than 255 characters. 13) The email has already been taken. 14) The password confirmation does not match.

Table 10. Admin Management module - authentication function.

ID	User group
AUTHENTICATION	Guest
Name	
Authentication	
Description	
Logging in to the system as admin by a guest	
Pre-conditions	
2) Not logged in the system	

Input
<ul style="list-style-type: none"> • User's email - up to 255 characters • User's password - up to 255 characters, minimum 8 characters • Remember me (optional)
Processing
<ol style="list-style-type: none"> 5) Check if email contains "@" in it <ol style="list-style-type: none"> a) if not, return error message (1) 6) Check if email is not longer than 255 characters <ol style="list-style-type: none"> a) if not, return error message (4) 7) Check password length <ol style="list-style-type: none"> a) if less than 8 characters, return error message (2) b) if longer than 255 characters, return error message (3) 8) Check if email and password matches record <ol style="list-style-type: none"> a) if does not, return error message(5)
Post-conditions
<ol style="list-style-type: none"> 3) Logged in the system 4) Admin role 5) Is not BLOCKED
Outputs
If the admin is successfully logged in, he sees his personal post page.
Error messages
<ol style="list-style-type: none"> 6) Please include '@' in the email address. 7) The password must be at least 8 characters. 8) The password may not be greater than 255 characters. 9) The email may not be greater than 255 characters. 10) These credentials do not match our records.

Table 11. Admin Management module - page customization function.

ID	User group
PAGE CUST	Admin
Name	
Page customization	
Description	
Customizing page layout.	
Pre-conditions	
4) Logged in the system 5) Admin role only 6) Opened page settings	
Input	
<ul style="list-style-type: none"> • Profile picture (optional) • Posts per page (optional) • Font size (optional) • Background color (optional) • Do not show post author (optional) • Sound on new comments (optional) • Sound on new notification (optional) 	
Processing	
2) Changes user page layout preferences in database	
Post-conditions	
2) Page gets refreshed and has its view updated	
Outputs	
Page with updated view based on selected preferences.	

Table 12. Admin Management module - permission function.

ID	User group
USER PERMISSIONS	Admin
Name	
Permissions	
Description	
Changes permissions of any selected User	
Pre-conditions	
4) Logged in to the system as Admin 5) Opened list of all users by View user list function (<i>table 6</i>) 6) Select function “Change permission” for selected user	
Input	
<ul style="list-style-type: none"> • Admin read permissions • Admin post permissions • Admin comment permissions • Admin delete post permissions • Admin block option • Admin ban option • All the above changes are available for the admin in relation to any user 	
Processing	
2) Update record field “user permissions” for the selected user in database, based on the selected permissions, chosen from drop-down list 3) Update own record as admin	
Post-conditions	
2) User permissions get updated in database 3) Admin records is updated in database	

Outputs
Returns a formatted list of all users of the system (by View user list function, table 6) with changed user permissions.

Table 13. Admin Management module - view user list function.

ID	User group
USER LIST	Admin
Name	
View user list	
Description	
Proceed and shows the list of all users of the system with corresponding user roles and permissions	
Pre-conditions	
3) Logged in to the system as Admin 4) Clicked on “View users” button	
Processing	
3) Selects all Users from database 4) Adds “Change role” (<i>table 7</i>) and “Change permissions” (<i>table 5</i>) function near each user	
Post-conditions	
2) Full user list and function shown	
Outputs	
Returns a formatted list of all users of the system, consisting of user’s name, email, role,	

permission and “Change role” function (*table 7*), and “Change permissions” (*table 5*). There always will be at least 1 user – admin itself.

Table 14. Admin Management module - user role changing function.

ID	User group
USER ROLES	Admin
Name	
User role changing	
Description	
Changes the role of the selected user to different to add pre-set permissions.	
Pre-conditions	
4) Logged in to the system as Admin 5) Opened list of all users by View user list function (<i>table 6</i>) 6) Select function “Change role” for selected user	
Input	
<ul style="list-style-type: none"> • User role 	
Processing	
2) Updates record field “user role” for the selected user in the database, based on the role chosen from the drop-down list.	
Post-conditions	
2) User role and permissions get updated in database	
Outputs	
Returns a formatted list of all users of the system (by View user list function, table 6) with	

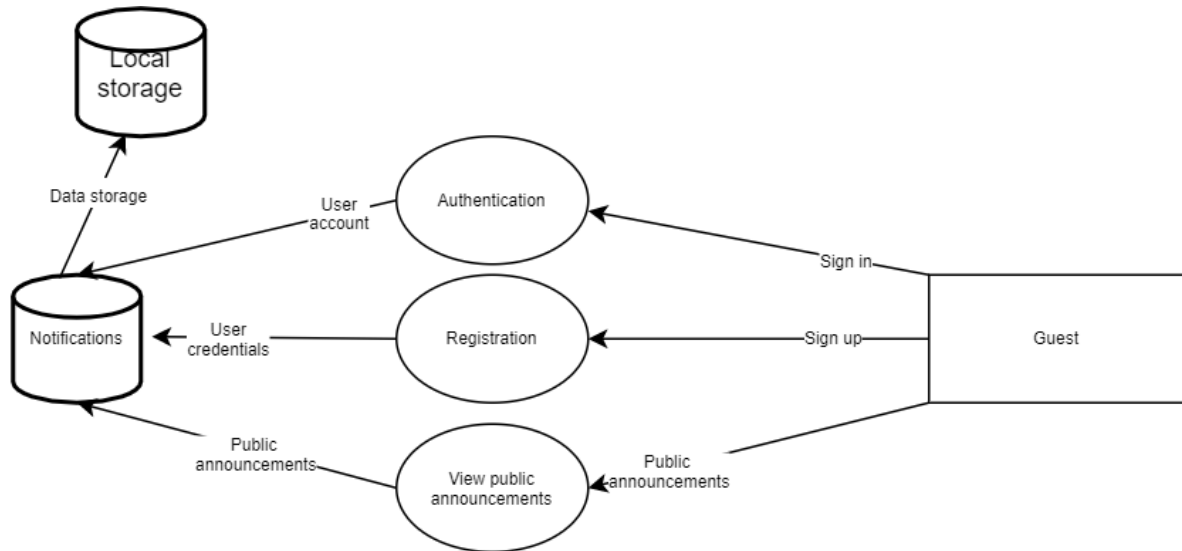
changed user role.

Table 15. Admin Management module - logout function.

ID	User group
LOGOUT	Admin
Name	
Logout	
Description	
Logout from the system by admin	
Pre-conditions	
3) Registered admin 4) Logged in to the system	
Input	
<ul style="list-style-type: none">Click “Logout” button	
Processing	
2. Log out from the system with session manipulation	
Post-conditions	
2) Logged out of the system	
Outputs	
Main page of the system.	

2.2.5. Guest management module

The data flow diagram of the guest management module (*pic. 7*) is based on the function described in *table 1*.



Pic. 7. Guest management module - data flow diagram (level 2)

Table 16. Guest Management module - registration function.

ID	User group
REGISTRATION	Guest
Name	
Registration	
Description	
New user registration by a guest	
Pre-conditions	
5) Not logged in the system 6) Account does not exist	
Input	

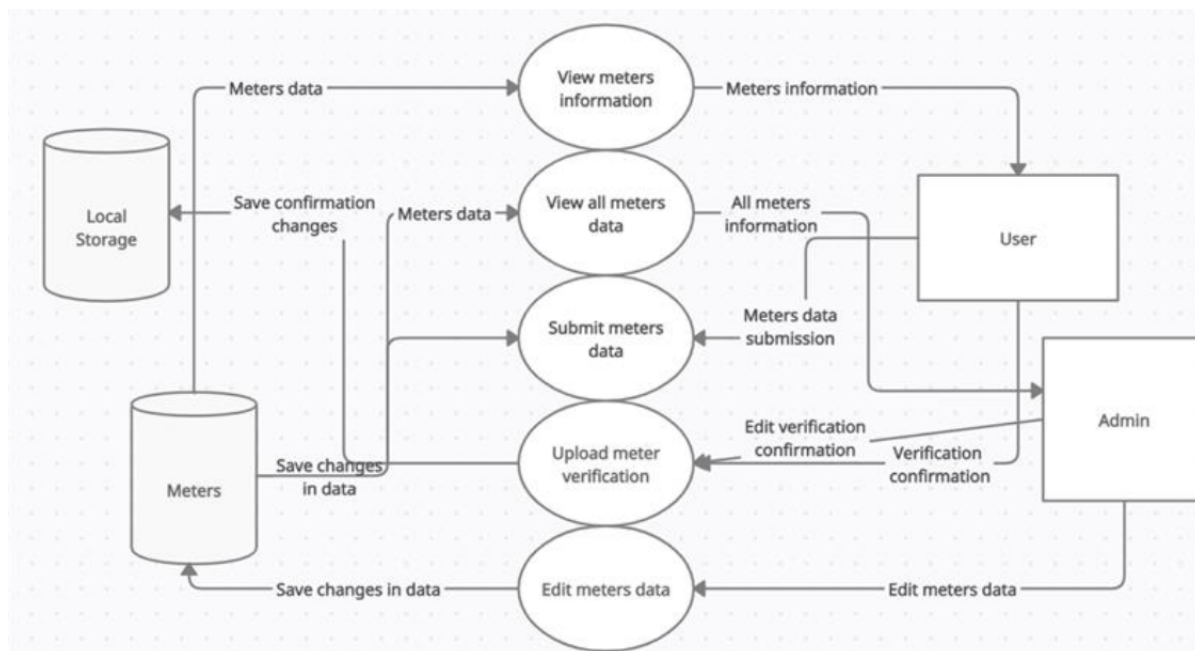
<ul style="list-style-type: none"> • User's name - up to 255 characters • User's email - up to 255 characters • User's password - up to 255 characters, minimum 8 characters
Processing
<p>15) Check if email contains "@" in it</p> <p> c) if not, return error message (1)</p> <p>16) Check if user's name is less than 255 characters</p> <p> a) if not, return error message (4)</p> <p>17) Check if email is not longer than 255 characters</p> <p> a) if not, return error message (5)</p> <p>18) Check if email is already registered</p> <p> a) if exists, return error message (6)</p> <p>19) Check password length</p> <p> a) if less than 8 characters, return error message (2)</p> <p> b) if longer than 255 characters, return error message (3)</p> <p>20) Check password confirmation field</p> <p> a) if is not same as password field, return error message (7)</p> <p>21) Add new user to database</p>
Post-conditions
<p>5) Registered in the system</p> <p>6) Logged in the system</p>
Outputs
If user is successfully added to the system, he will be automatically logged in the system
Error messages
<p>15) Please include '@' in the email address.</p> <p>16) The password must be at least 8 characters.</p> <p>17) The password may not be greater than 255 characters.</p> <p>18) The name may not be greater than 255 characters.</p> <p>19) The email may not be greater than 255 characters.</p> <p>20) The email has already been taken.</p> <p>21) The password confirmation does not match.</p>

Table 17.. Guest Management module - permission function.

ID	User group
USER PERMISSIONS	Guest
Name	
Permissions	
Description	
Permissions for Guest users	
Pre-conditions	
7) Not a registered user	
Input	
<ul style="list-style-type: none"> • Guest public post read permissions • Guest register permissions 	
Processing	
4) Creating a user account while guest 5) Reading any public post available	
Post-conditions	
4) User permissions get updated in database	
Outputs	
Only if the user is registered the system saves all the information in the database about new user (not a Guest anymore, see registered user's tables for more info)	

2.2.6. Data Submission module

The data flow diagram of the Data Submission module *pic.8* is based on the functions described in table 1. Each function is separately described in *tables 18-22*.



Pic.8. Data submission module - data flow diagram (level 2)

Table 18. Data Submission module – View meters information

ID	User group
METR INFO	User
Name	
View meters information	
Description	
Allow to users to view information about meters	

Pre-conditions
1. Logged to the system as User
Processing
1. Checks if user sign-in <ul style="list-style-type: none"> a. if not, then error message (1) 2. If login is correct – profile is visible
Output
Returns sign in data, where is visible your profile information (table)
Error messages
(1) You are not Sign-in

Table 19. Data Submission module – View all meters data

ID	User group
ALL METR INFO	Admin
Name	
View all meters data	

Description
Allows admin to view all information about user profiles
Pre-conditions
1. Logged to the system as Admin
Processing
1. Retrieves all meters data submissions for selected as quantaty profiles
Output
Returns a formatted data about meters information associated with users profiles and uploading verification reports with each meters entry (table 21).

Table 20. Data Submission Module – Submit meters data

ID	User group
SUBM METR INFO	User
Name	
Submit meters data	
Description	
Allow to users watch others profiles information	

Pre-conditions
<ol style="list-style-type: none"> 1. Logged to the system as User 2. Opened “view meters information” (<i>table 18</i>).
Input
<ul style="list-style-type: none"> · All fields are filled with data collected from meters
Processing
<ol style="list-style-type: none"> 1. Checks if user have profile <ol style="list-style-type: none"> a. if not, user not found , then error message (1) 2. If user is found – collect present necessary data from meters <ol style="list-style-type: none"> a. if wrong data type or blank, then error message (2) b. if blank, then error message (3)
Output
Submits a formatted data about user profiles to the database and returns “view meters information” (<i>table 13</i>) with new data
Error messages
<ol style="list-style-type: none"> (1) No such profile (2) Show similar profiles (3) New data could not be shown as empty field

Table 21. Data Submission Module – Upload meters verification

ID	User group
UPD METR VERIF	User, Admin
Name	
Upload meters verification	

Description
Allow to users and admins to upload meters verification report
Pre-conditions
<ol style="list-style-type: none"> 1. Logged to the system as User 2. Opened “view meters information” (<i>table 18</i>) or <ol style="list-style-type: none"> 1. Logged to the system as Admin 2. Opened “view all meters data” (<i>table 19</i>)
Input
· Verification report upload
Processing

<p>If User:</p> <ol style="list-style-type: none"> Checks user profiles <ol style="list-style-type: none"> if user not found profile, show error message (1) If profile is found – select meter to which upload verification Uploads verification report to local storage <ol style="list-style-type: none"> if left blank or wrong document type, then error message (2) <p>If Admin:</p> <ol style="list-style-type: none"> Selects “upload verification” for meter in “view all meters data” (<i>table 14</i>) Uploads verification report to local storage <ol style="list-style-type: none"> if left blank or wrong document type, then error message (2)
Output
<p>If User</p> <ul style="list-style-type: none"> Returns “View meters information” (<i>table 18</i>) <p>If Admin:</p> <ul style="list-style-type: none"> Returns “View all meters data” (<i>table 19</i>)
Error messages
<ol style="list-style-type: none"> No such profile Wrong upload

Table 22. Data Submission Module – Edit meters data

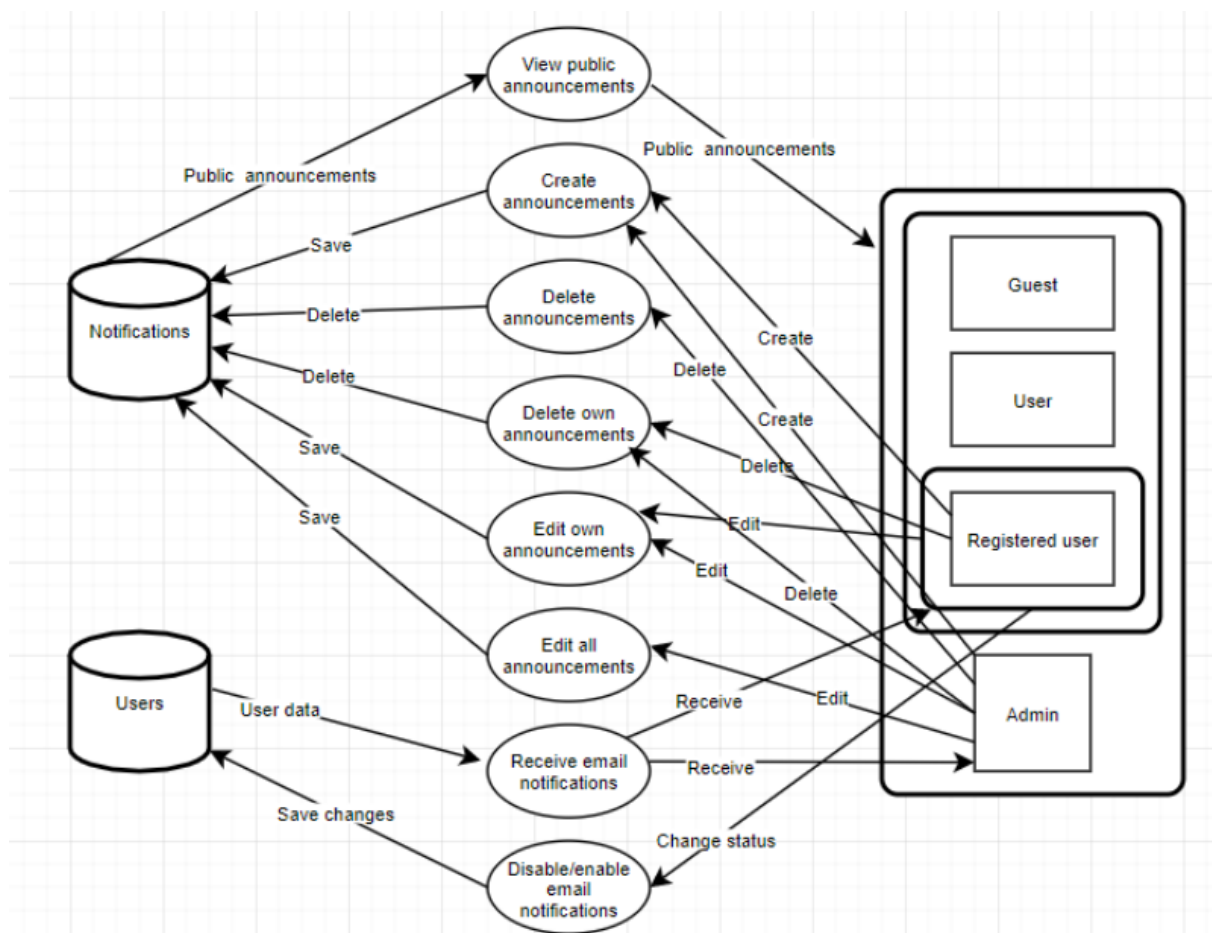
ID	User group
EDIT METR DATA	Admin
Name	
Edit meters data	

Description
Edition of meters data
Pre-conditions
<ol style="list-style-type: none"> 1. Logged to the system as Admin 2. Opened list of all user's profiles view all meter's data" (<i>table 19</i>). 3. Select function "Edit meters data" for a specific profile's meter
Processing
<ol style="list-style-type: none"> 1. Validates all input fields (meter data, verified till) <ol style="list-style-type: none"> a. if wrong data type, blank or incorrect format, then error message (1) 2. Updates record field "user's profile" of the selected user in the database
Output
Returns a formatted list of meters data (by "view all meters data" function, <i>table 14</i>).

Error messages
(1) Wrong data

2.2.7. Notifications module

Notifications module is separated per function, which is based on functions described in *table 1*.



Pic.9. Notifications module - data flow diagram (level 2)

Table 23. Notification module - public view function.

ID	User group
PUBLIC VIEW	Guest, User, Admin
Name	
Public view	
Description	
Allows to see all public posts made in the system	
Pre-conditions	
1) Systems web page with public posts is opened 2) User is not Blocked	
Processing	
1) Selects all public posts from the database	
Post-conditions	
1) Public posts are visible to the requestor	
Outputs	
Returns all public posts in a preformatted way.	

Table 24. Notification module - private view function.

ID	User group
PRIVATE VIEW	User, Admin

Name
Private view
Description
Allows to see all private posts made in the system
Pre-conditions
1) Logged in to the system as User or Admin 2) User is not Blocked
Processing
1) Selects all private posts from the database
Post-conditions
1) Private posts are visible to the requestor
Outputs
Returns all privateposts in a preformatted way.

Table 25. Notification module - create note function.

ID	User group
CREATE NOTE	User, Admin
Name	
Create note	
Description	
Create a new note in the system.	

Pre-conditions
<ol style="list-style-type: none"> 1) Logged in to the system 2) User is not Blocked 3) Select “create new note”
Input
<ul style="list-style-type: none"> • Header text • Message text • Note type • Attachments (optional)
Processing
<ol style="list-style-type: none"> 1. Validate all mandatory fields <ol style="list-style-type: none"> a. if field left blank, then output error (1)
Post-conditions
<ol style="list-style-type: none"> 1) New note is added to the database
Outputs
If user is successfully added to the system, he will be automatically logged in the system
Error messages
<ol style="list-style-type: none"> 1) All mandatory fields have to be filled

Table 26. Notification module - delete note function.

ID	User group
DELETE NOTE	Admin
Name	

Delete note
Description
Allows to delete any note in the system.
Pre-conditions
1) Logged in to the system as Admin 2) Opened list of all note
Input
<ul style="list-style-type: none"> Select note/-s to delete
Processing
1) Deletes selected note/-s from the database
Outputs
Returns to the main page of the systems website

Table 27. Notification module - delete private note function.

ID	User group
DELETE PRIVATE NOTE	User, Admin
Name	
Delete private note	
Description	
Deleting own note in the system	
Pre-conditions	
1) Logged in to the system as User or Admin	

2) User is not Blocked 3) Open list of all personal notes
Input
<ul style="list-style-type: none"> Select note/-s to delete
Processing
1) Deletes selected note from the database
Outputs
Returns to the main page of the systems website

Table 28. Notification module - edit private note function.

ID	User group
EDIT PRIVATE NOTE	User, Admin
Name	
Edit private note	
Description	
Edit own note in the system	
Pre-conditions	
1) Logged in to the system as User or Admin 2) User is not Blocked 3) Open list of all personal notes	
Input	
<ul style="list-style-type: none"> Select note to edit Change data of fields (header text, message text, note type, attachments) 	

Processing
<ol style="list-style-type: none"> 1. Validate all mandatory fields <ol style="list-style-type: none"> a. if field left blank, then output error (1)
Post-conditions
1) Note is edited in the database
Outputs
Returns to the main page of the systems website
Error messages
1) All mandatory fields have to be filled

Table 29. Notification module - edit public note function.

ID	User group
EDIT PUBLIC NOTE	Admin
Name	
Edit public note	
Description	
Edit any note in the system	
Pre-conditions	
<ol style="list-style-type: none"> 1) Logged in to the system as Admin 2) Open list of all public notes 	
Input	
<ul style="list-style-type: none"> • Select note to edit • Change data of fields (header text, message text, note type, attachments) 	

Processing
<ol style="list-style-type: none"> 1. Validate all mandatory fields <ol style="list-style-type: none"> a. if field left blank, then output error (1)
Outputs
Returns to the main page of the systems website
Error messages
<ol style="list-style-type: none"> 1) All mandatory fields have to be filled

Table 30. Notification module - receive notifications about new note function.

ID	User group
RECEIVE NOTIFICATION	User, Admin
Name	
Receive notification about new note	
Description	
Send notification when a new note is created	
Pre-conditions	
<ol style="list-style-type: none"> 1) Logged in to the system as User or Admin 2) User is not Blocked 3) Notifications are not disabled (<i>table 9</i>) 	
Input	
<ul style="list-style-type: none"> • Notification message data 	
Processing	
<ol style="list-style-type: none"> 1) Check if notifications are not disabled for the user 	

a) if are disabled, function is terminated for the user 2) Composes the notification message 3) Sends the notification
Outputs
Notification of new note is sent to the user

Table 31. Notification module - disable or enable notifications function.

ID	User group
RECEIVE NOTIFICATIONS	User
Name	
Disable/enable notifications	
Description	
Ability to disable or enable notifications	
Pre-conditions	
1) Logged in to the system as User or Admin 2) User is not Blocked 3) Opened user settings	
Input	
<ul style="list-style-type: none"> Select email notifications status (on/off) 	
Processing	
1. Updates user data in the database based on the selected option	
Outputs	
Success message on the same page	

2.3. Non-functional requirements

For the system described following non-functional requirements:

- The application design should be user friendly to provide the best convenience for mobile and web devices.
- Application has to be accessible from all browsers, for example: Safari, Chrome, Mozilla, Edge.
- Application needs to require users to create accounts to access user private profiles. Application will grant access to accounts when users enter the correct username and password.
- Application may not grant access until the user creates a strong password.
- All user passwords need to be saved in the system encoded way.
- The system interface should be provided for English, Latvian and Russian languages.
- Data processing speed must be fast in order to provide a response to the user. Speed also involves assessing a system's ability to manage an increasing workload as you use different applications at the same time.
- Clean the database once a year.

3. Software design description

3.1. Database design

Based on the functional description and conceptual database model (*pic.3*) logical database was created (*pic. 10*).

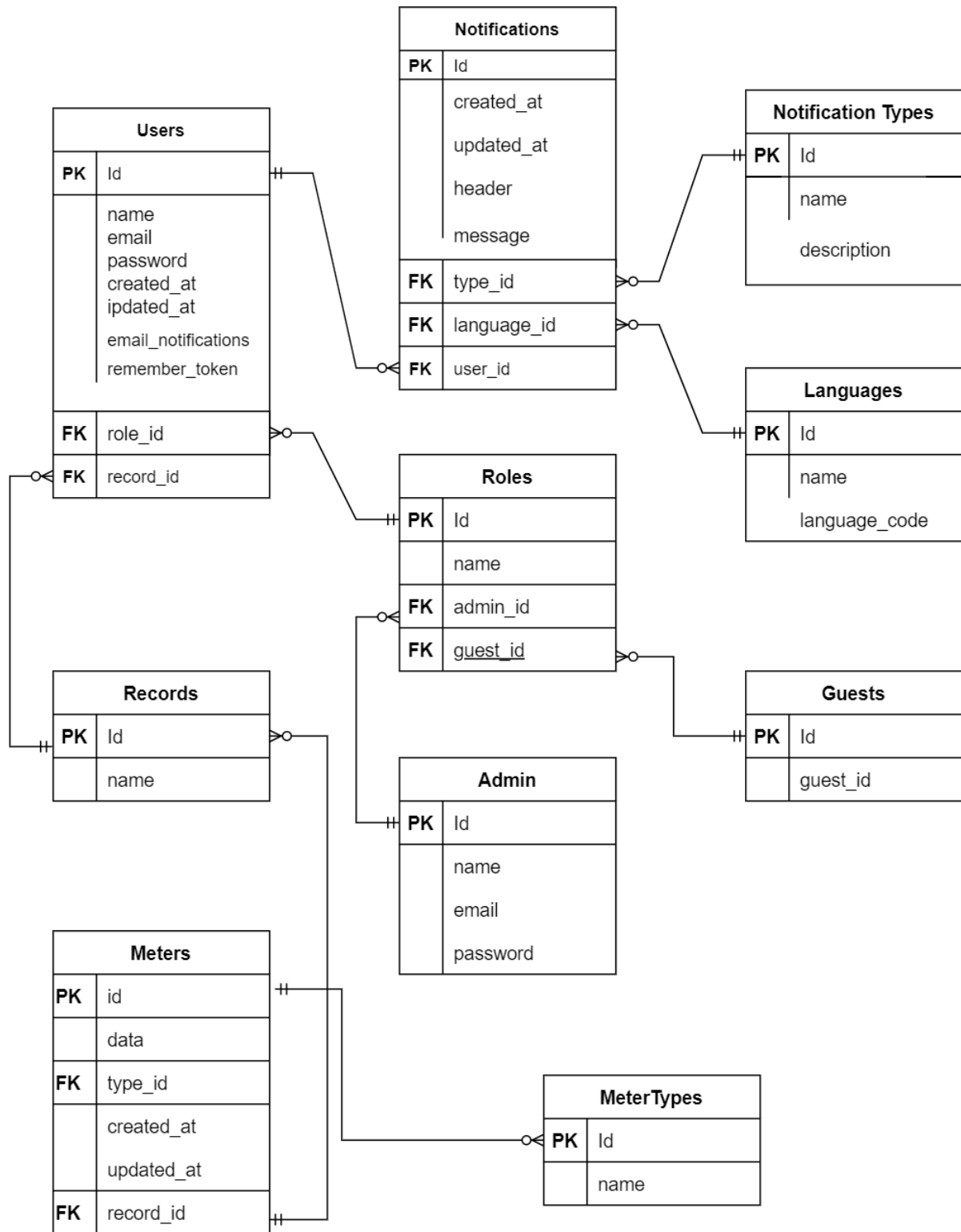
Based on the logical model, the physical database model was created. It includes the same number of entities as the logical database model but includes main nullable and datatypes columns as well (*pic. 11*).

The terms of data types and other applicable parameters to database content may vary depending on the used DB. Was used in terms of Microsoft SQL server.

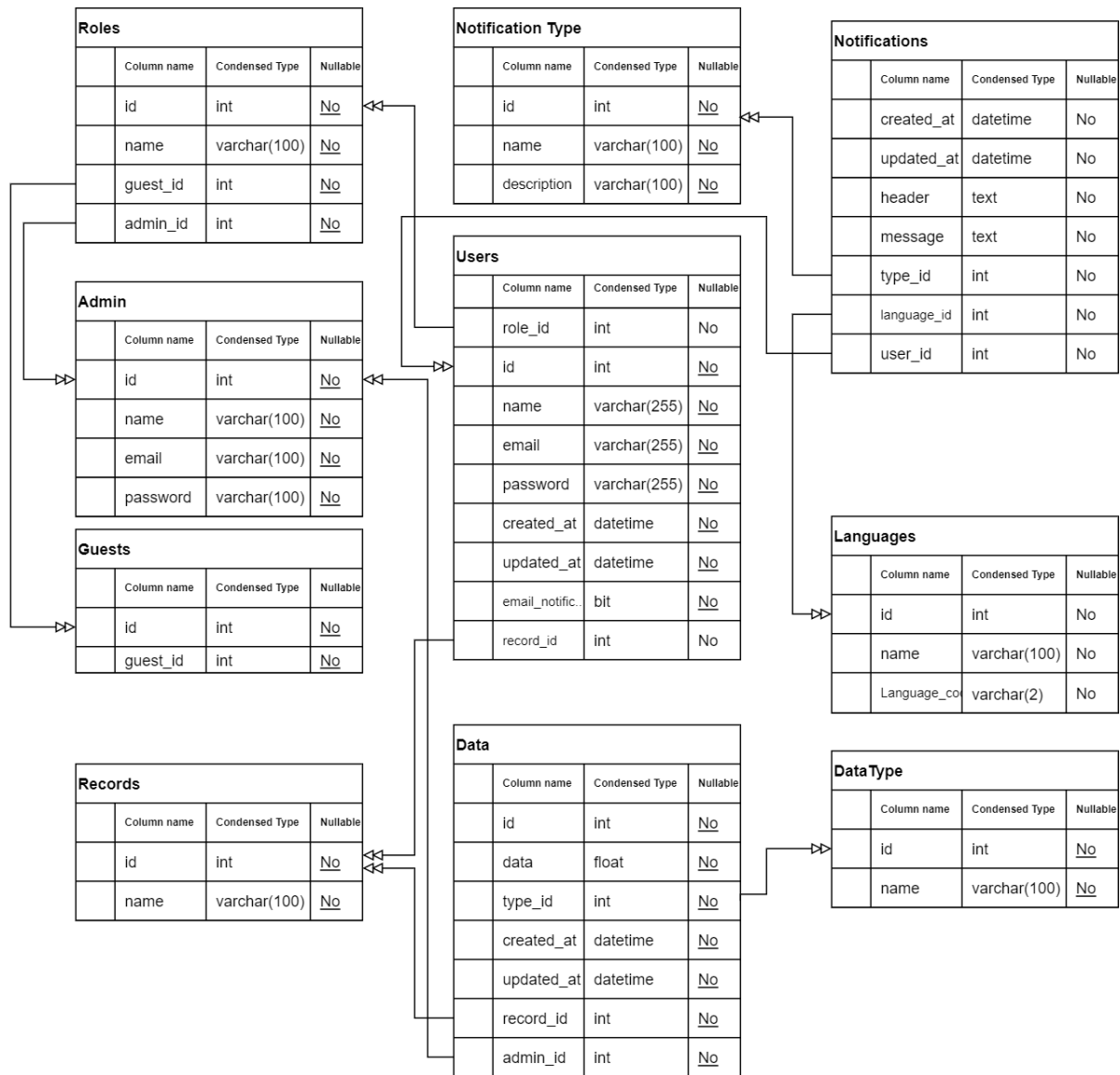
The description of database tables is divided into sections according to the modules shown in section 2.2.2.

- In subsection 3.1.1. described Users and corresponding entities.
- In subsection 3.1.2. described Admins and corresponding entities.
- In subsection 3.1.3. described Guests and corresponding entities.
- In subsection 3.1.5. described Data and corresponding entities.

Initial seedings examples are shown in the Appendix according to assumptions and dependencies described in the section 1.7.



Pic.10. Logical database model



Pic.11. Physical database model

3.1.1. User and corresponding entities

This subsection contains detailed information about table design of Users entity (*table 32*). Mainly the data from the Users entity is used by User Management module functions (*section 2.2.3., tables 2-8*).

Table 32. Users entity description

Key	Column name	Data type	Nullable	Other parameters	Description
PK	ID	int	No	IDENTITY(ID for user

				1,1)	identification
	name	varchar(255)	No		Used username
	email	varchar(255)	No	Unique	email address of user
	password	varchar(255)	No		user password to access system
	created_at	datetime	No		Date of creation account
	updated_at	datetime	No		Date of updated information to user account
	email_notifications	bit	No		Datatype which shows user notifications is on or off
	remember_token	varchar(255)	Yes		Token for “remember” field of login. Can be NULL if was not selected
FK	role_id	int	No	FK referenced from entity Roles (<i>table 33, ID attribute</i>)	The role of the user in the system
FK	record_id	int	NO	FK references from entity Record (<i>table 34, ID attribute</i>)	The record in system

User roles are displayed in the Roles table. The system users roles shown in section 1.5. of the document.

Table 33. Roles entity description

Key	Column name	Data type	Nullable	Other parameters	Description
PK	ID	int	NO	IDENTITY(1,1)	ID for user role identification
	name	varchar(100)	NO		Role name used in the system

Table 34. Records entity description

Key	Column name	Data type	Nullable	Other parameters	Description
PK	ID	int	NO	IDENTITY(1,1)	ID for records identification
	name	varchar(100)	NO		Records used in the system

3.1.2. Admins and corresponding entities

This subsection contains detailed information about table design of Admin entity (*table 35*). Admin entity is used in Admin management module functions (*section 2.2.4., tables 9-15*).

Table 35. Admin entity description

Key	Column name	Data type	Nullable	Other parameters	Description
PK	id	int	NO	IDENTITY(1,1)	ID of the admin
	name	varchar(100)	NO		Admin username

	email	varchar(100)	NO	Unique	Admin email
	password	varchar(100)	NO		Admin password

3.1.3. Guests and corresponding entities

This subsection contains detailed information about table design of Guests entity (*table 36*). Guest entity is used in Guest management module functions (*section 2.2.5., tables 16, 17*).

Table 36. Guests entity description

Key	Column name	Data type	Nullable	Other parameters	Description
PK	id	int	NO	IDENTITY(1,1)	ID for the user
	name	varchar(100)	NO		Used name for Guest

3.1.4. Data and corresponding entities

This subsection contains detailed information about table design of Data and its corresponding entities (*tables 37, 38*). Data entity is used in Data Submission module functions (*section 2.2.6., tables 18-22*).

Table 37. Data entity description

Key	Column name	Data type	Nullable	Other parameters	Description
PK	id	int	NO	IDENTITY(1,1)	Data ID
	data	float	NO		
FK	type_id	int	NO	FK referenced from entity DataType (<i>table 38, ID attribute</i>)	ID of the DataType
	created_at	datetime	NO		Data creation

					date
	updated_at	datetime	NO		Date when Data was last updated
FK	record_id	int	NO	FK referenced from entity Records (<i>table 34, ID attribute</i>)	The record in system
FK	admin_id	int	NO	FK referenced from entity Admin (<i>table 35, ID attribute</i>)	ID of the admin

Data has 3 corresponding entities - DataType (*table 38*), Records (*table 34*), Admin (*table 35*) entities. Below there is the description of the DataType entity only, as other entities are described above.

Table 38. DataType entity description

Key	Column name	Data type	Nullable	Other parameters	Description
PK	id	int	NO	IDENTITY(1,1)	ID for the data type
	name	varchar(100)	NO		Name of the data type

3.1.5. Notifications and corresponding entities

Notifications entity is used in Notifications module functions (*section 2.2.7., tables 23-25*).

Table 39. Notifications and corresponding entity description

Key	Column name	Data type	Nullable	Other parameters	Description
PK	id	int	No	IDENTITY (1, 1)	Artificial id for notification entry identification

	created_at	datetime	No		Date of creation of the corresponding notification entry
	updated_at	datetime	No		Date of information update of the corresponding notification entry
	header	text	No		Header of notification
	message	text	No		Body message of notification
FK	type_id			FK referenced from entity Notification Types (table 37, id attribute)	Type of notification
FK	language_id			FK referenced from entity Languages (table 38, id attribute)	Language of notification
FK	user_id			FK referenced from entity Users (table 32, id attribute)	Creator of notification

NotificationTypes entity and Languages entity (*tables 40, 41*) are codifiers tables to Notifications entity (table 38).

Table 40. NotificationTypes entity – description

Key	Column name	Data type	Nullable	Other parameters	Description
PK	id	int	No	IDENTITY (1, 1)	Artificial id for type of the notification

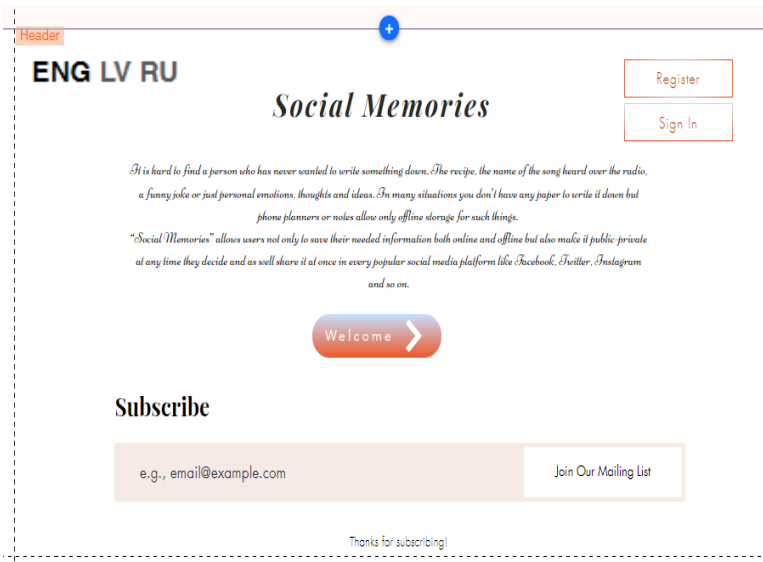
	name	varchar(100)	No		Name of the type
	description	varchar(100)	No		Description of the type

Table 41. Languages entity – description

Key	Column name	Data type	Nullable	Other parameters	Description
PK	id	int	No	IDENTITY (1, 1)	Artificial id for language identification
	name	varchar(100)	No		Name of the language
	language_code	varchar(2)	No		Language code according to ISO 639-1:2002 standard [1]

3.2. Partial Design of User Interfaces

This section contains the mockups of the platform pages, it's functional and platform design overview for both PC and Smartphone users. The Homepage (pic. Main Page) contains main information about the platform, “Register” button which allows users to register, “Sign In” button for entering as registered user, “Welcome” button which redirects to the page with all the public posts visible for guests as well as a subscription option. In each page there is a language change, which is a non-functional requirement described in section 2.3.



Social Memories

It is hard to find a person who has never wanted to write something down. The recipe, the name of the song heard over the radio, a funny joke or just personal emotions, thoughts and ideas. In many situations you don't have any paper to write it down but phone planners or notes allow only offline storage for such things.

"Social Memories" allows users not only to save their needed information both online and offline but also make it public-private at any time they decide and as well share it at once in every popular social media platform like Facebook, Twitter, Instagram and so on.

Sign In

Register

Welcome

Subscribe

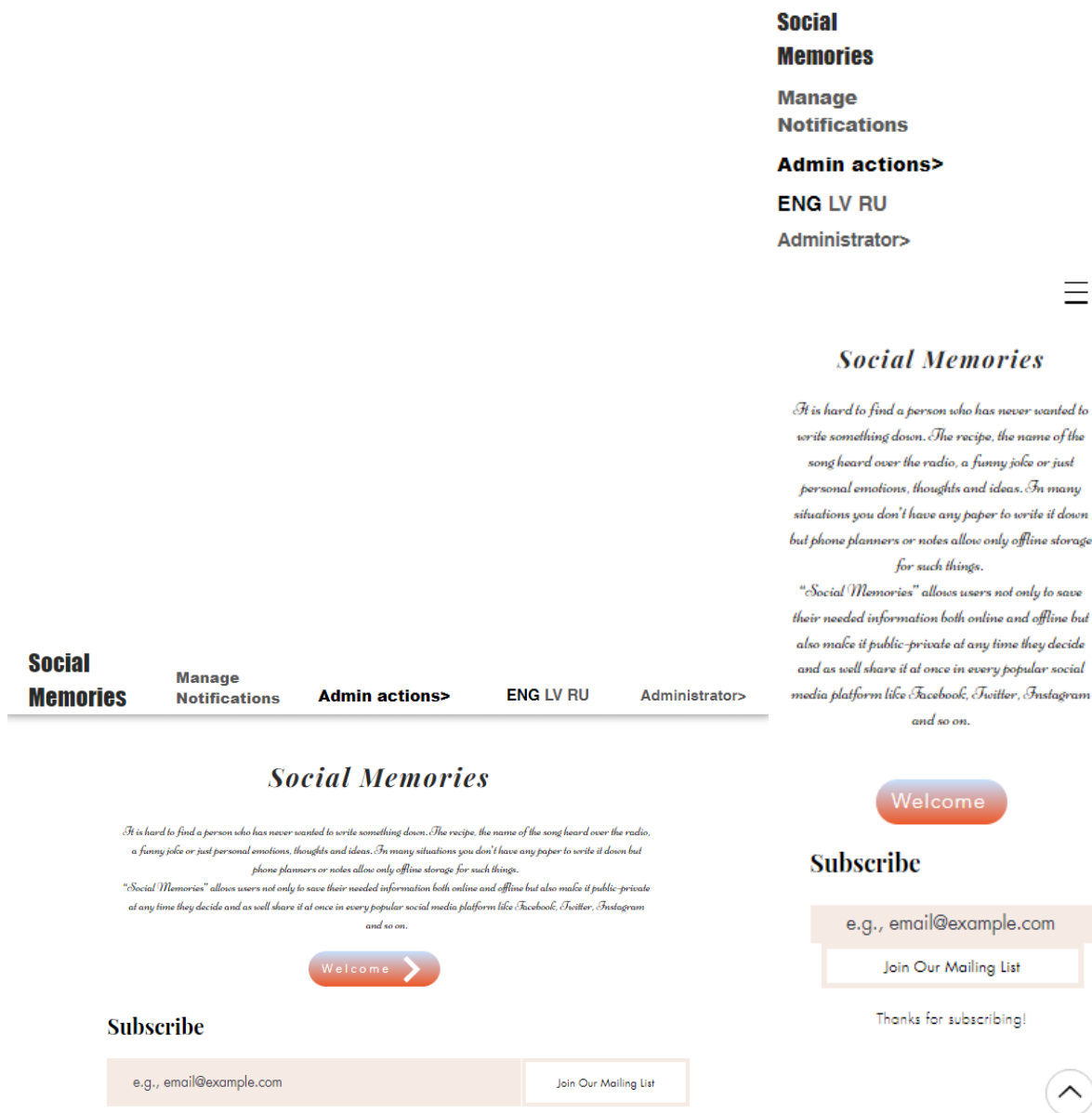
e.g., email@example.com

Join Our Mailing List

Thanks for subscribing!



Pic.12. Main page (Guest)



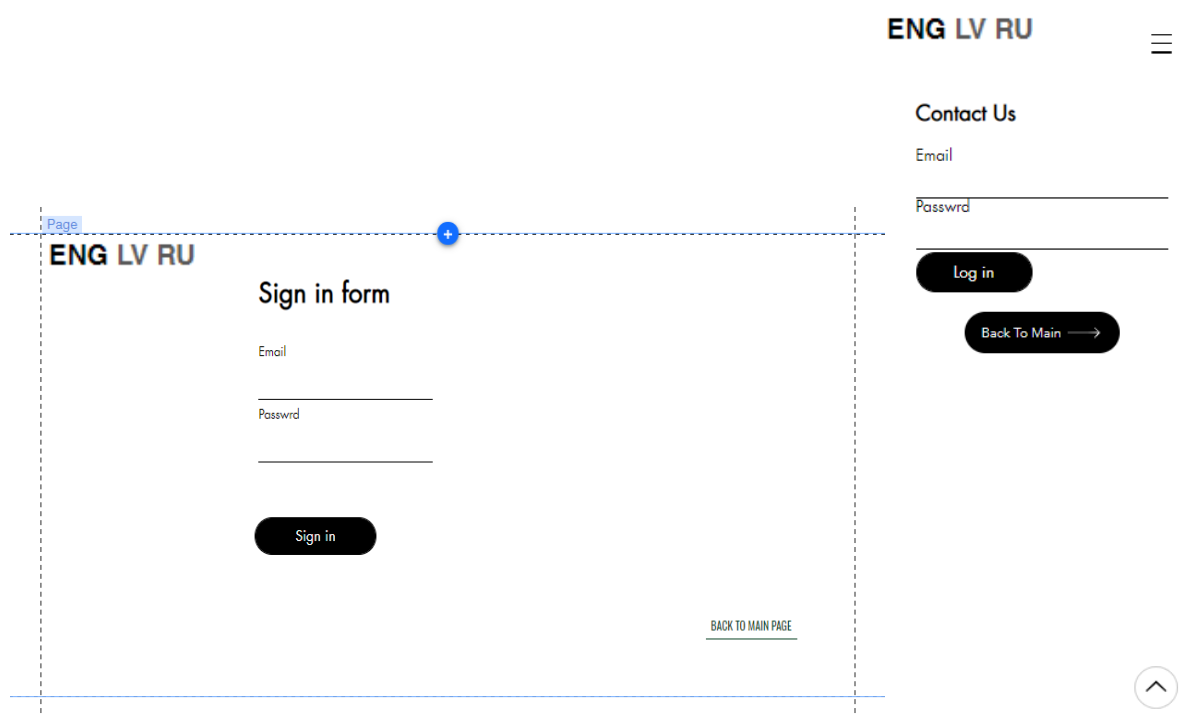
Pic.13. Main page (Admin)

Registration form (*pic 14. Registration form*) includes fields required by functional requirement in sections 2.2.3-2.2.5. After the registration form is filled in, the user has to click the “Register” button in order to successfully finish the registration process. If a user changes his mind, he can press the “Back to Main Page” button and return to the Home Page.

The image displays two versions of a registration form. The desktop version on the left features a header with language options (ENG, LV, RU) and a title 'REGISTRATION'. It includes input fields for First Name, Last Name, Email (marked with an asterisk), Password (marked with an asterisk and a blue thumbs-up icon), and Re-Password (marked with an asterisk and a blue thumbs-up icon). A black 'Register' button is positioned below the fields. At the bottom, there is an orange 'Back To Main' button with a right-pointing arrow. The mobile version on the right has a similar header and title. It includes input fields for First Name, Last Name, Email (marked with an asterisk), Password (marked with an asterisk and a blue thumbs-up icon), and Re-Password (marked with an asterisk and a blue thumbs-up icon). A black 'Register' button is located below the fields, followed by the text 'Thanks for registering!'. At the bottom, there is an orange 'Back To Main' button with a right-pointing arrow and a circular button with an upward-pointing arrow.

Pic.14. Registration form

When a user is a registered one and uses the “Sign In” option, the following page (*pic 15. Sign in form*) will be shown for him asking to put his email address and password for signing up to the platform. This is a functional requirement which can be found in sections 2.2.3-2.2.5.



The image shows a wireframe of a 'Sign in form' page. The page is divided into two main sections by a vertical dashed line. On the left, the header 'ENG LV RU' is at the top left. Below it, the title 'Sign in form' is centered. Under the title, there are two input fields labeled 'Email' and 'Passwrd'. Below these fields is a black button with the text 'Sign in'. At the bottom right of this section, there is a link that says 'BACK TO MAIN PAGE'. On the right side of the page, the header 'ENG LV RU' is at the top right, followed by a hamburger menu icon. Below the header, there is a section titled 'Contact Us' with two input fields labeled 'Email' and 'Passwrd'. Below these fields are two black buttons: 'Log in' and 'Back To Main' with a right-pointing arrow. At the bottom right of the page, there is a circular button with an upward-pointing arrow.

Pic.14. Sign in form

SOCIAL MEMORIES

Manage Notifications

ENG LV RU

User>

Create new notification

Notification No1

Notification's author (self); Date; Type

Notification's text

Edit notification

Delete notification

Notification No2

Notification's author (self); Date; Type

Notification's text

Edit notification

Delete notification

SOCIAL MEMORIES

Manage Notifications

ENG LV RU

User>

≡

Create new notification

Notification No1

Notification's author (self); Date; Type

Notification's text

Edit notification

Delete notification

Notification No2

Notification's author (self); Date; Type

Notification's text

Edit notification

Delete notification

⬆

Pic.16. Notification (User)

The admin notification module (*pic 16. Notifications*) is a functional requirement which can be seen in section 2.2.7. The page allows an admin to create, edit, delete any notification connected info of any user as well as return back to the Main page using the “Back to Main” button.

Social Memories **Manage Notifications** **Admin actions>** **ENG LV RU** **Administrator>**

Create new notification

Notification No1

Notification's author (self); Date; Type

Notification's text

Edit notification Delete notification

Notification No2

Notification's author (self); Date; Type

Notification's text

Edit notification Delete notification

Back To Main →

Social Memories

Manage Notifications

Admin actions>

ENG LV RU

Administrator>

Create new notification

Notification No1

Notification's author (self); Date; Type

Notification's text

Edit notification Delete notification

Notification No2

Notification's author (self); Date; Type

Notification's text

Edit notification Delete notification

Back To Main →

Pic.16. Notifications (Admin)

Change role option page (*pic. 17 User role change*) accessible only for an admin. Can be found in functional requirements section 2.2.4. Admin can choose any role for the particular user from the drop-down and click the “Save role” button for changing it.

The image displays the 'Change user role' page in the 'Social Memories' application. The page is divided into three main sections: a header, a main content area, and a sidebar.

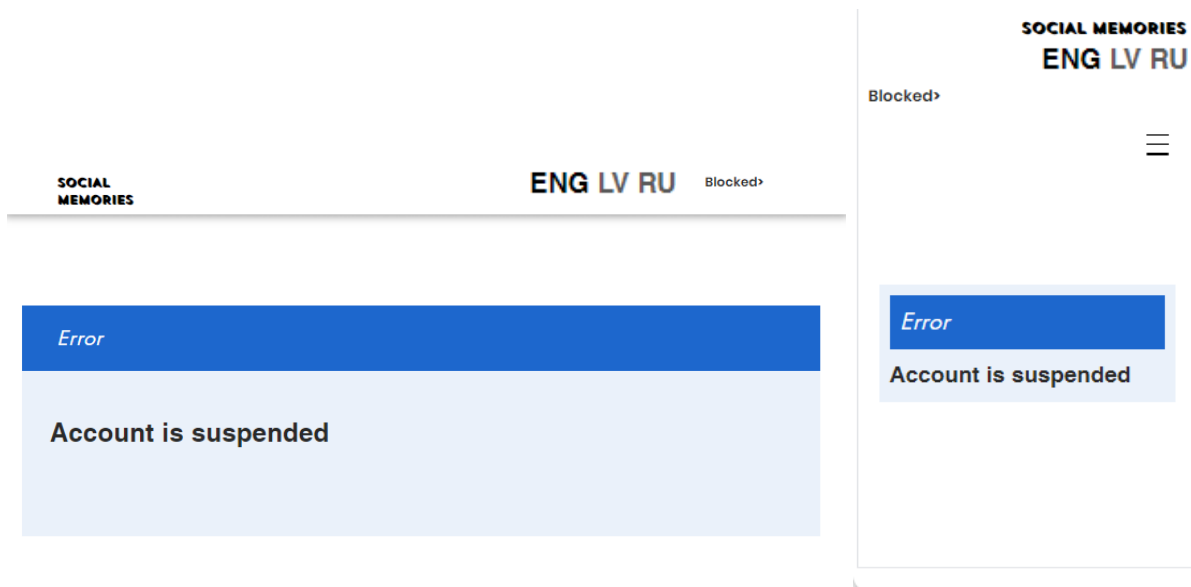
Header: The header contains the application name 'Social Memories' on the left. In the center, there are navigation links: 'Manage Notifications', 'Admin actions>' (which is highlighted with a blue underline), 'ENG LV RU', and 'Administrator>'. On the right side of the header, there is a language selector 'ENG LV RU' and a user profile section showing 'Administrator>'.

Main Content Area: The main content area has a green header 'Change user role'. Below this, there are labels for 'Username:' and 'User email:'. A form box contains a dropdown menu labeled 'Select role' and a blue 'Save role' button. At the bottom right of the main content area, there is a green button labeled 'Back To Main' with a right-pointing arrow.

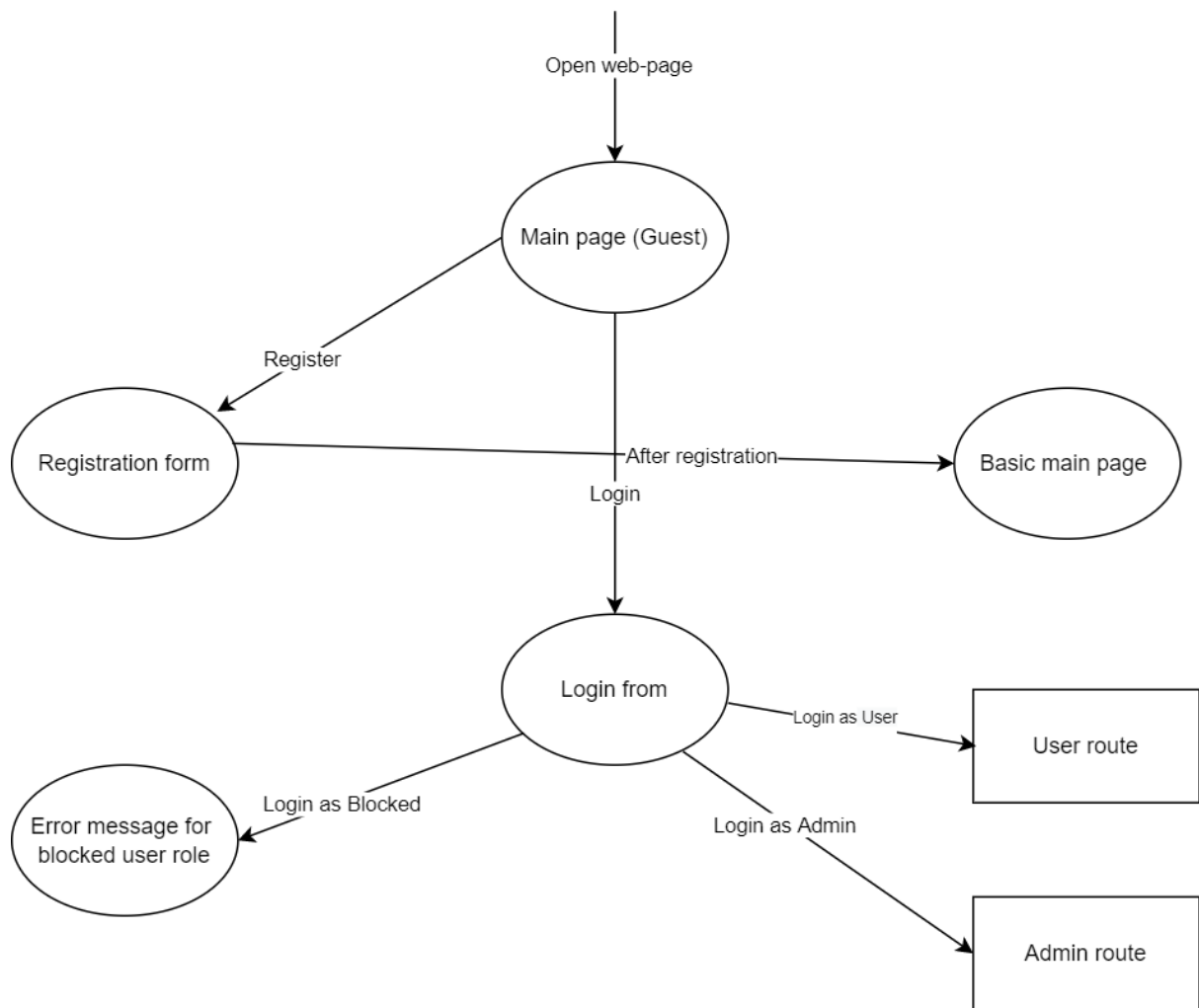
Sidebar: The sidebar is located on the right side of the page. It has a green header 'Change user role'. Below this, there are labels for 'Username:' and 'User email:'. A form box contains a dropdown menu labeled 'Select role' and a blue 'Save role' button. At the bottom of the sidebar, there is a green button labeled 'Back To Main' with a right-pointing arrow.

Pic.17. User role change (Admin)

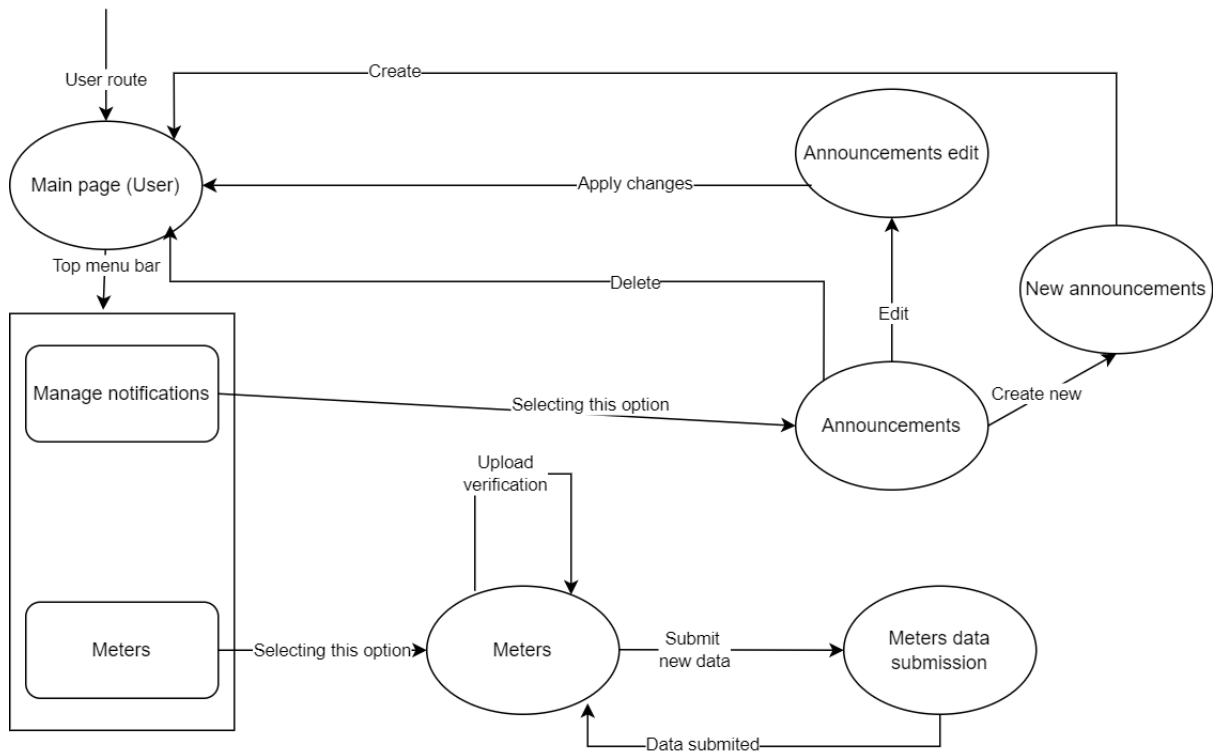
The block option page (*pic.18. Blocked*) can be seen by the blocked user after an admin has changed the role of that user. Can be seen in sections 2.2.3-2.2.4 of the functional requirements. User will not be able to open his or other users' posts, comment, like or do anything on the platform if he was blocked by an admin.



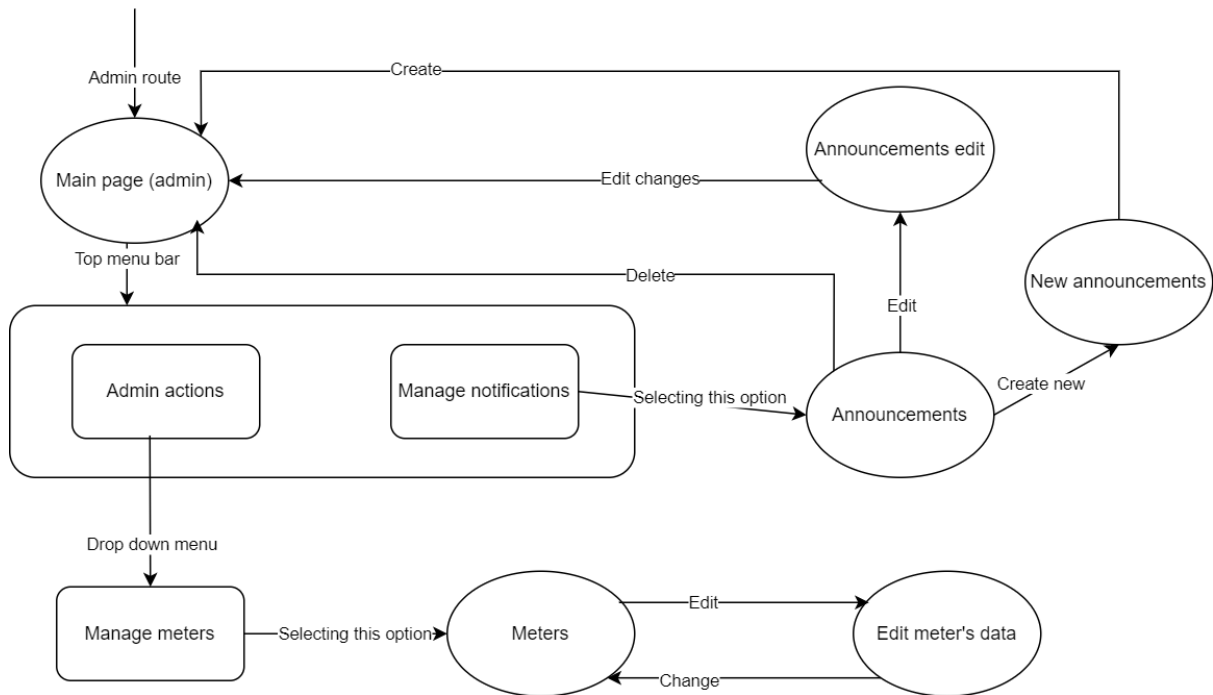
Pic.18. Blocked



Pic.19 User interface diagram (overall)



Pic.20 User interface diagram



Pic.21 Admin interface diagram

3.3. Partial Functional Design

The section contains diagrams, which describes the most important part of the functionality of the system.

- The process of configuring an account for User role is shown in (pic.22). It's based on section 2.2.3. (table 2), section 2.2.5 (tables 16, 17).

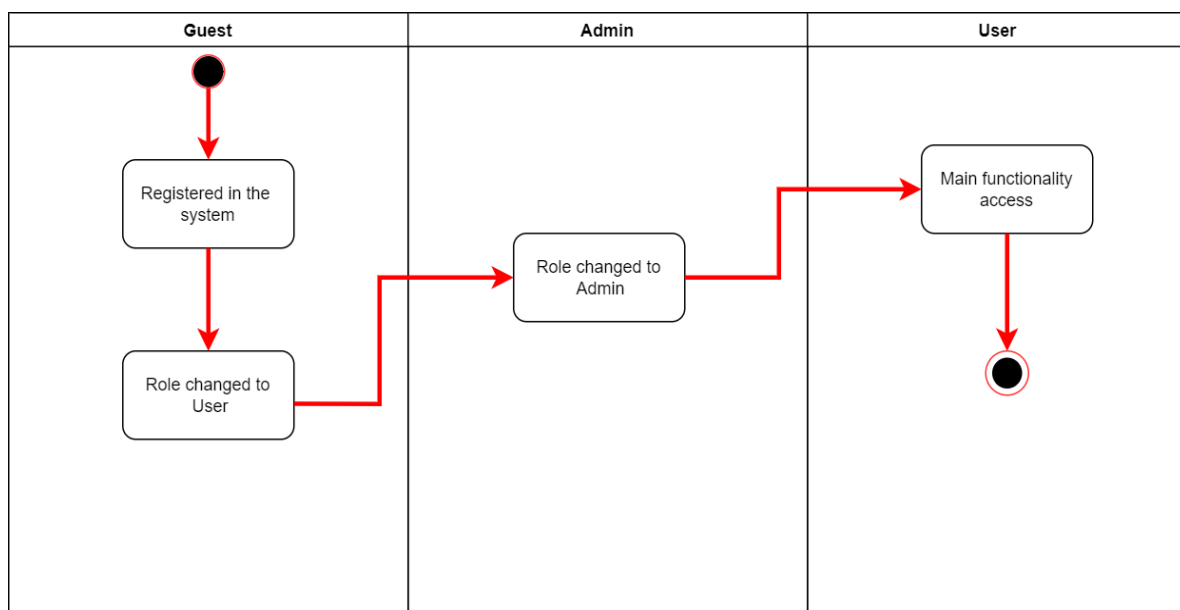
- The main parts of the system are shown in the Pic.23. It utilizes MVC approach, as well as shows possible necessary side packages and local storage. Inner framework files and configuration are not shown. It combines all sections described in the document: functional description (sections 2.2.3.-2.2.7.), database structure (section 3.1.) in the form of models (even though the database itself is not shown explicitly) and GUI, described in section 3.2. as views.

- Process of login is shown in the Pic.24. It is based on function in table 3 of section 2.2.3

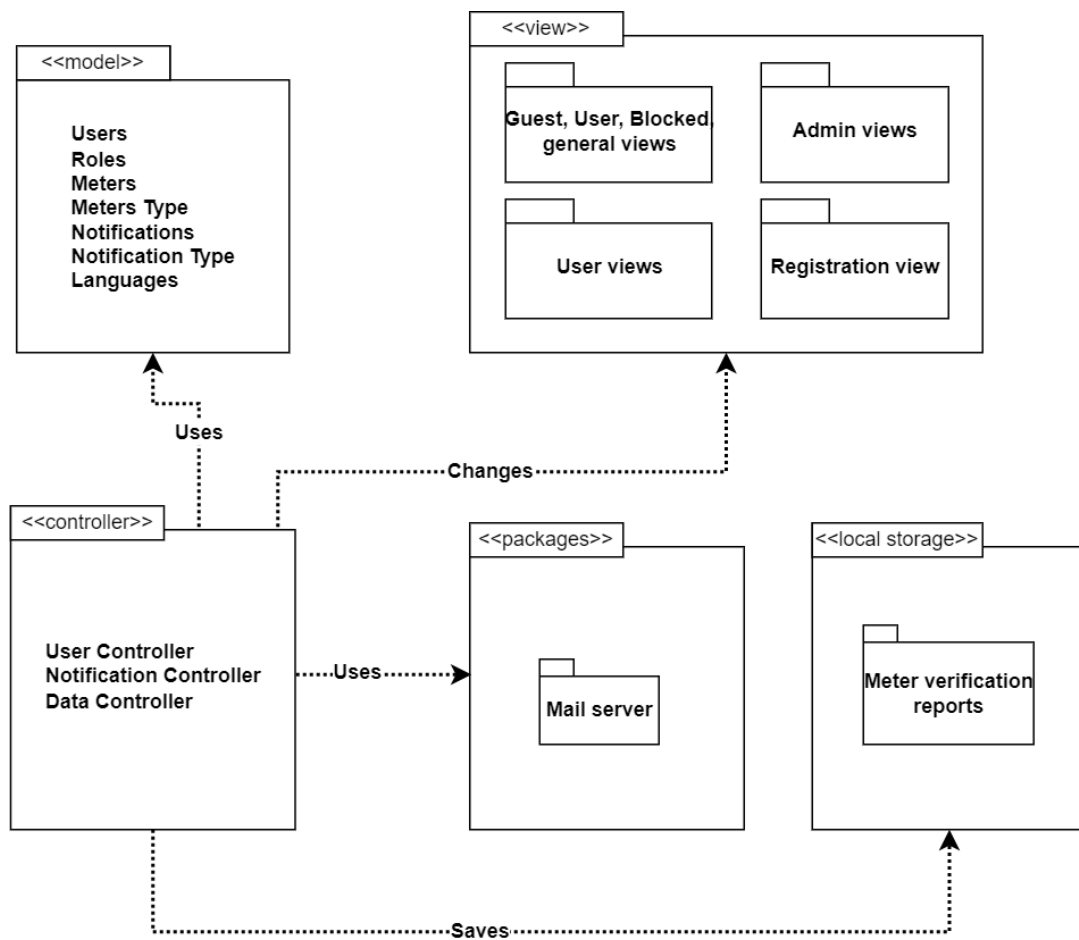
- Process of user registration is shown in the Pic.25. It is based on the function in table 2 of section 2.2.3.

Record status change (both manual and automatic) is shown in Pic. 26. It is mostly based on section 2.2.7 and its corresponding tables 23-31.

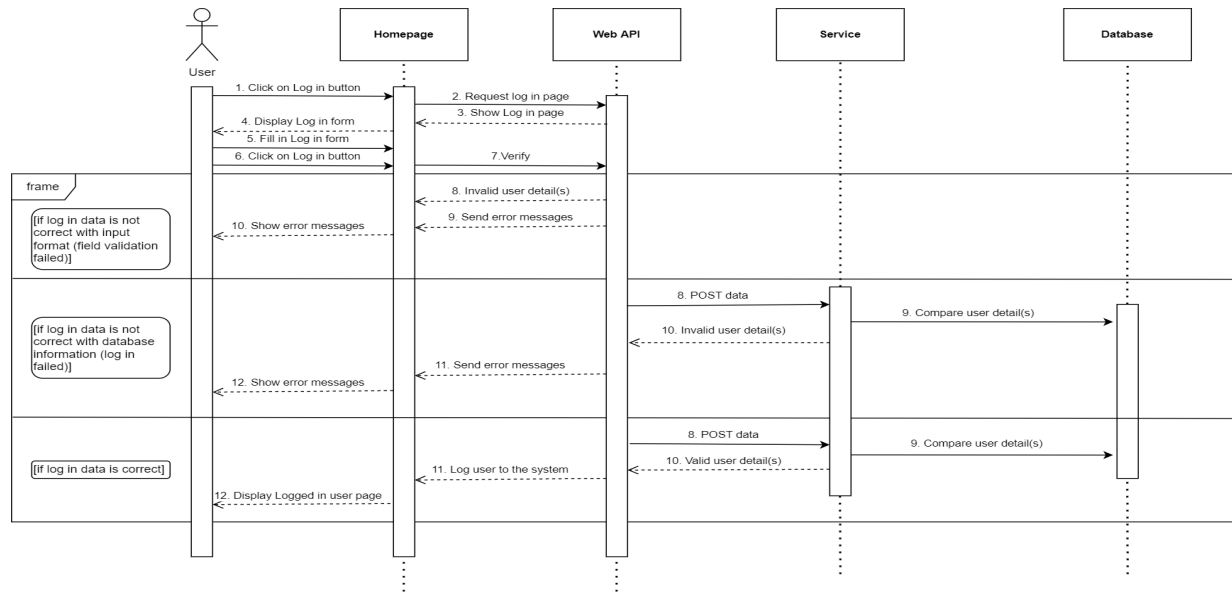
Class Diagram description is in a form of MVC as shown in Pic. 27. Main centerpiece being Roles (controller), where it is meant Admin function described in Table 14 in section 2.2.4. And Notification controller is meant to be all functions shown in Pic 9. of section 2.2.7. Model classes are corresponding database entities from section 3.1.



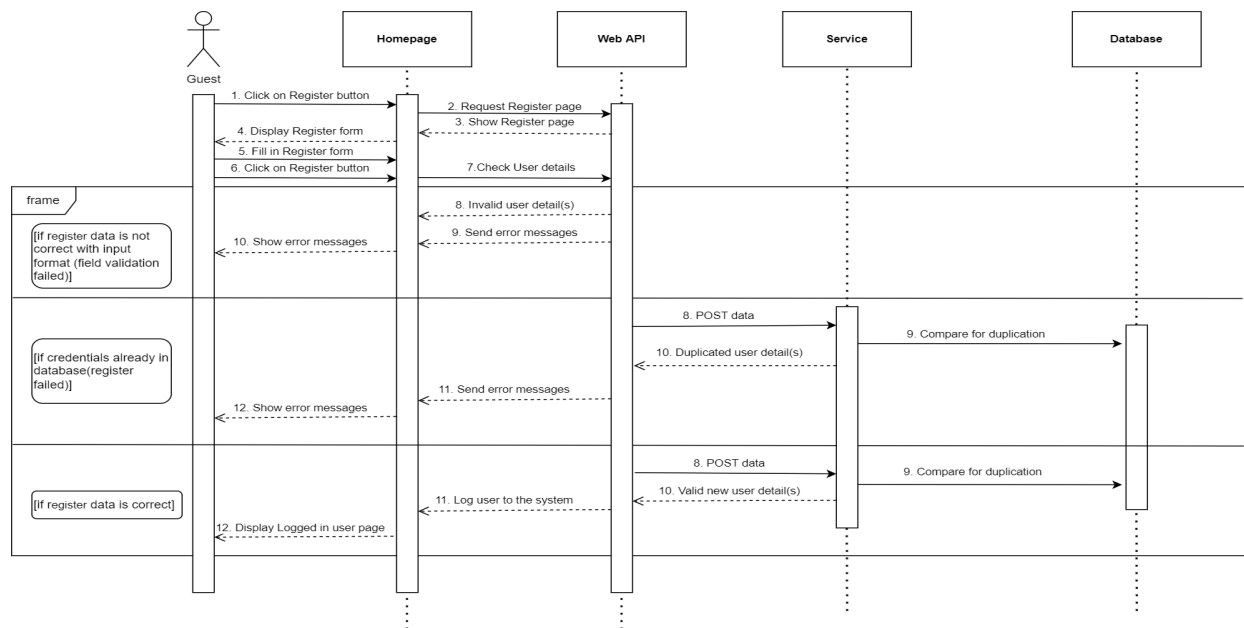
Pic.22 Configuring working account (UML activity diagram)



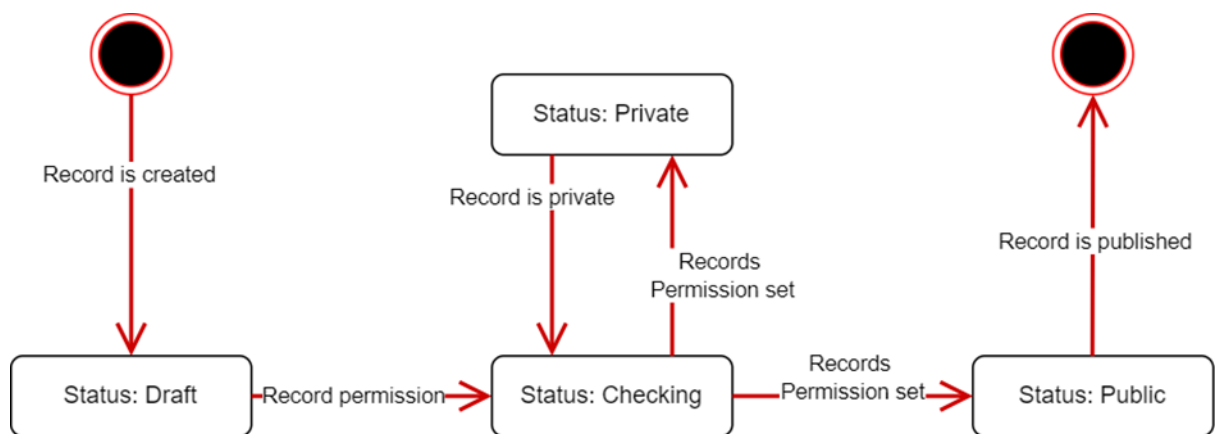
Pic.23 Main part of system (UML package diagram)



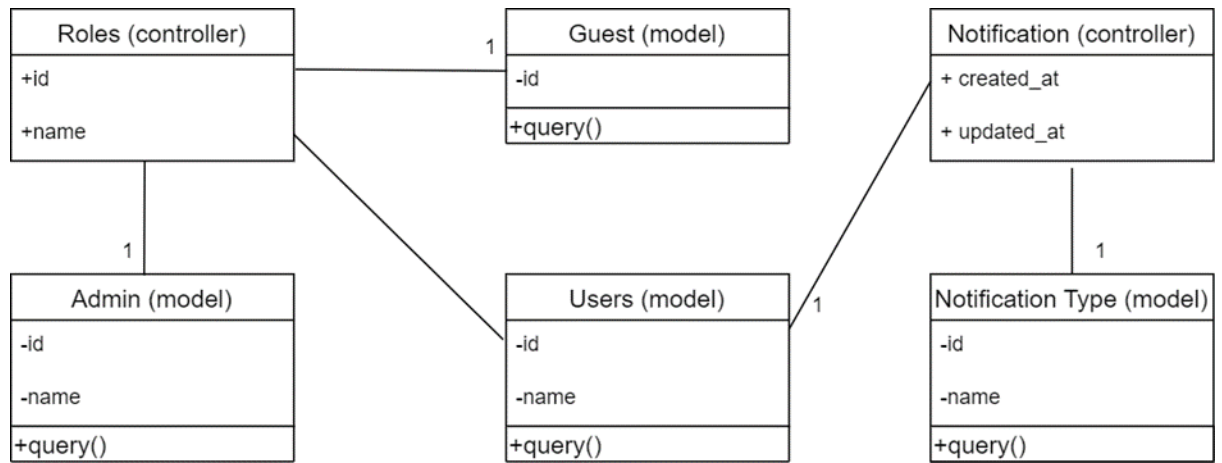
Pic.24 Log in process (UML sequence diagram)



Pic.25 Registration process (UML sequence diagram)



Pic.26 Status change



Pic.27 UML Class Diagram

References and sources

1. “ISO/IEC/IEEE International Standard - Systems and software engineering -- Life cycle processes -- Requirements engineering,” in ISO/IEC/IEEE 29148:2018(E), pp. 1- 104, 30 Nov. 2018, DOI: 10.1109/IEEESTD.2018.85596