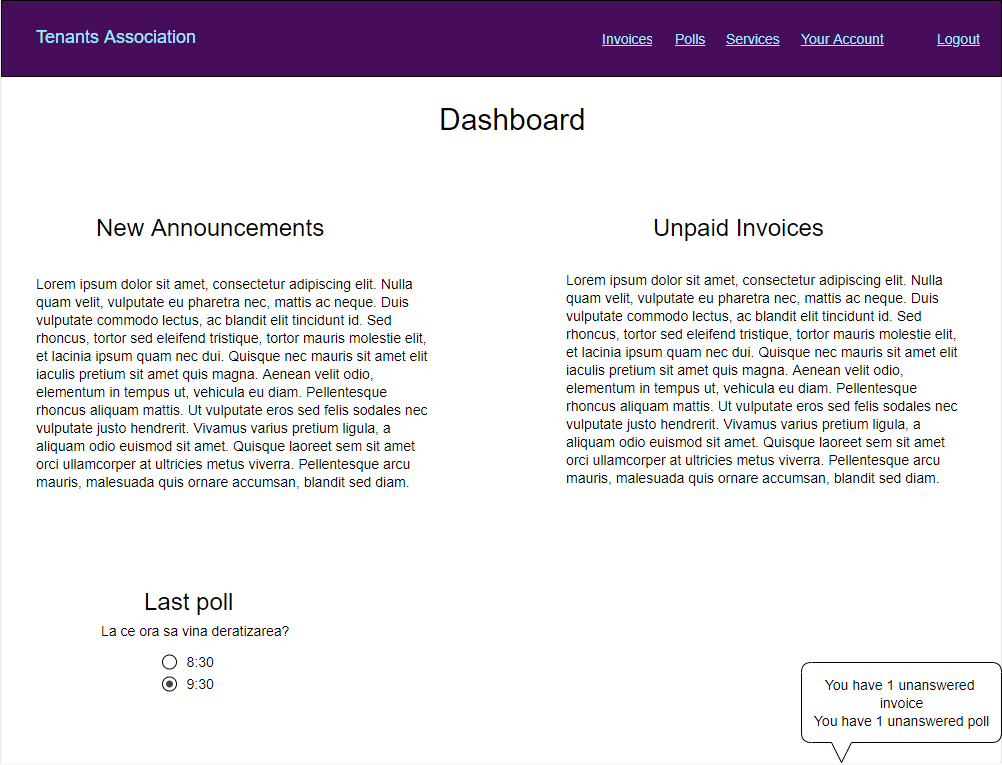
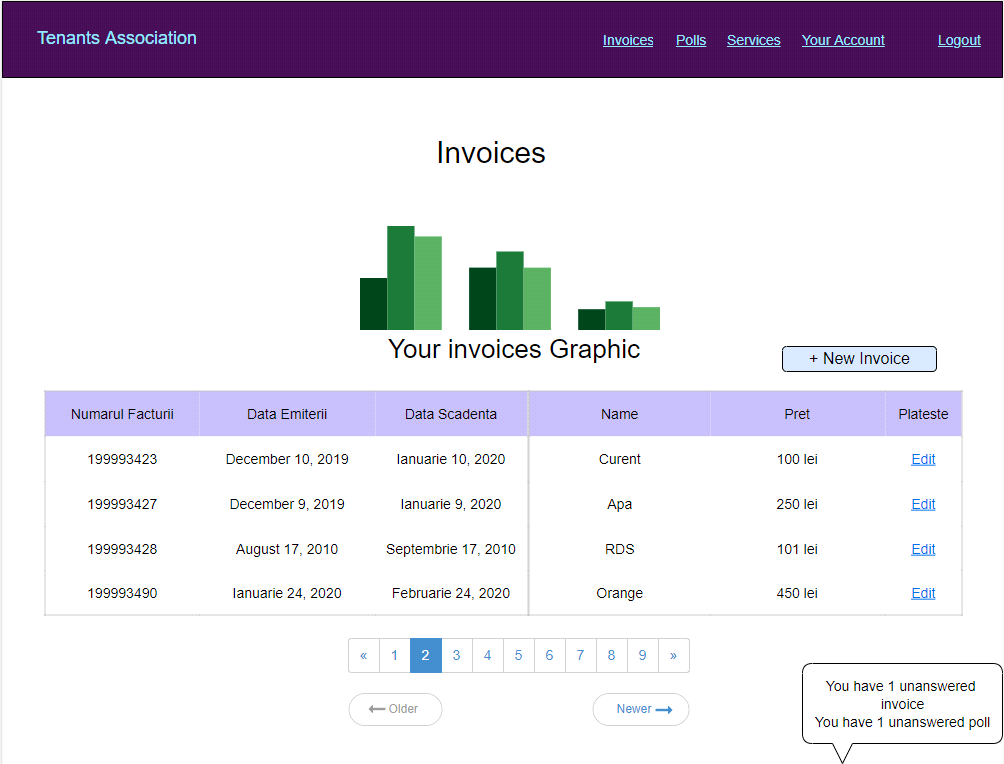
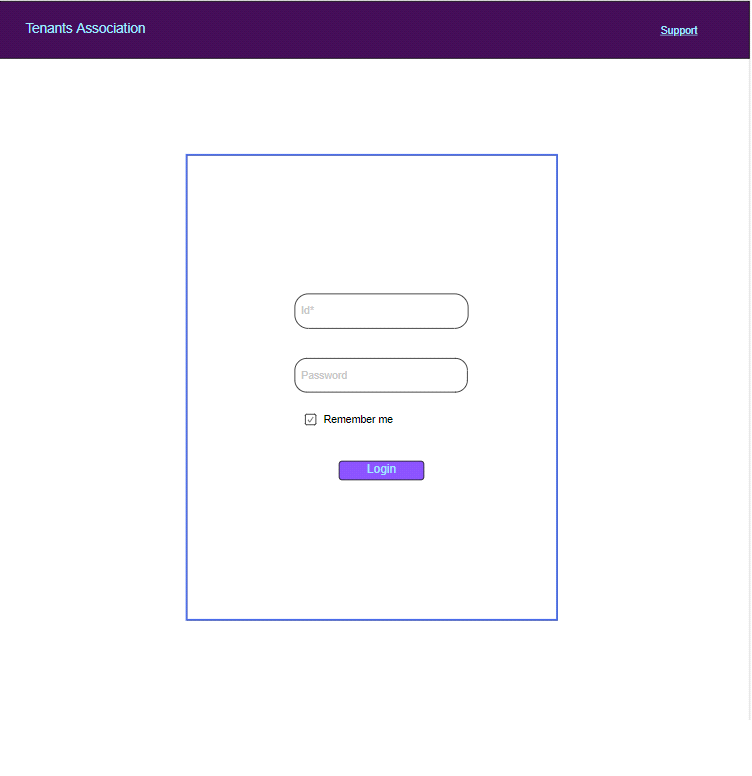
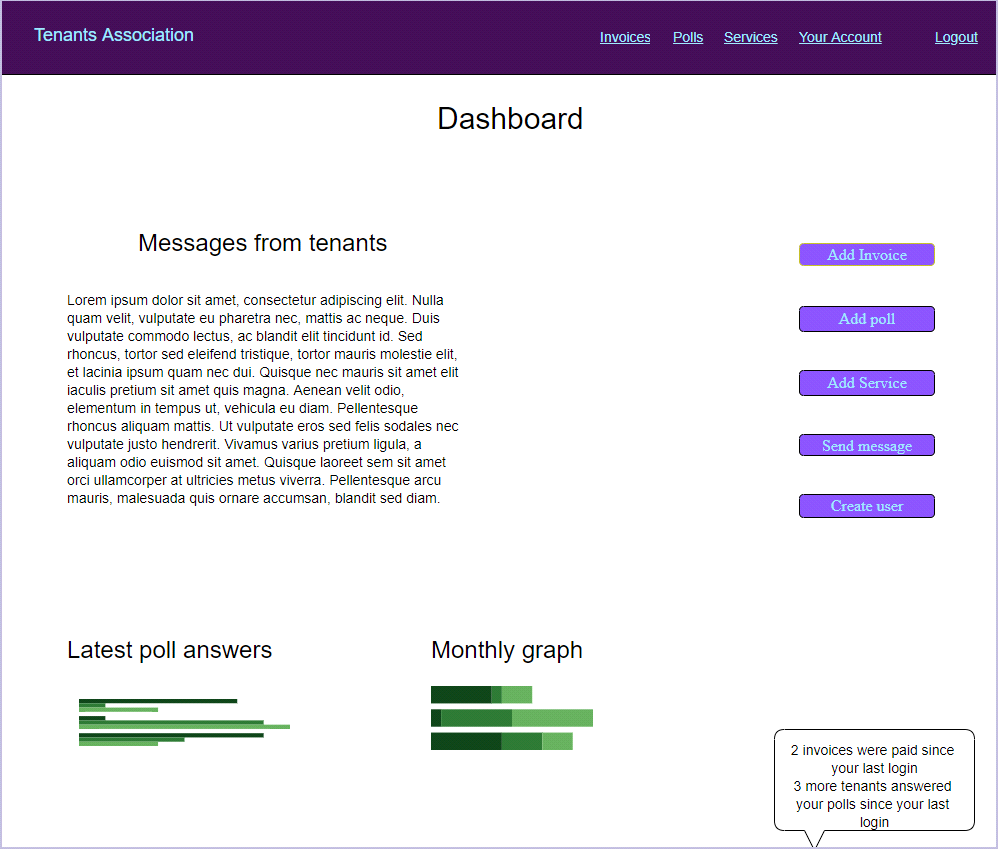
**3. External Interface Requirements**

**3.1 User Interfaces**

The layout should be as follows:

* + - The header which displays:
      * the logo
      * the currently logged user
      * the “Logout” and “Your Account” menu
    - The body, which displays
      * the menu, immediately under the header
      * the content(already created)



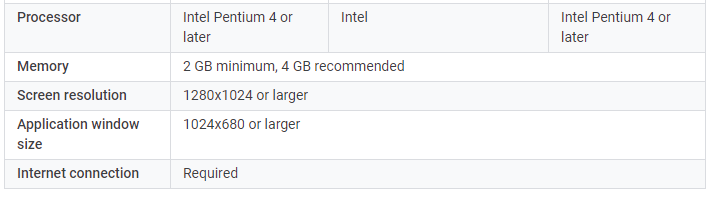


There will be two different user interfaces that will accompany this website: one for the users and one for the administrators.

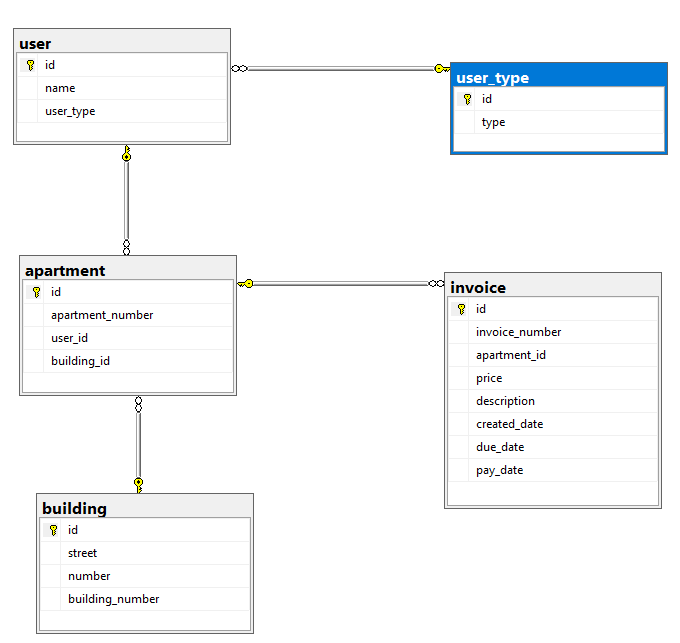
The administrators will be able to add users and remove them, to add bills , to create pools etc.

The users will be able to see the bills, answer to the pools created by the administrators etc.

**3.2 Hardware Interfaces**



Both x86 and x64 architectures.

**3.3 Software Interfaces**

The application is connected to a database (SQL Server). The database will communicate with the server, and the server will send the information to the client application. This a mock-up picture of the

database.

**3.4 Communications Interfaces**

HTTP and HTTPS will be the communication protocols that our app will use.

**4. System Features**

* **User Class 1 – Regular user**
  1. **Functional Requirement 1**

ID: 1

TITLE: Login in the website

DESC: An user should be able to login with an user defined by the administrator

DEP: 4

**1.2 Functional Requirement 2**

ID: 2

TITLE: Logout from the website

DESC: The user should be able to logout from the website

DEP: 1

* 1. **Functional Requirement 3**

ID: 3

Title: User modify personal data

DESC: The user can modify it’s password, location etc.

DEP: 1

* **User Class 2 – Administrator**

**2.1 Functional Requirement 1**

ID: 4

TITLE: Administrator login

DESC: The administrator will have a special account with more permissions than the regular user

DEP: None

**2.2 Functional Requirement 2**

ID: 5

TITLE: Administrator add new user

DESC: The administrator can add a new user(tenant) everytime someone moves in the building.

DEP: 4

**2.3 Functional Requirement 3**

ID: 6

TITLE: Administrator edit user

DESC: The administrator can edit user data(password, address, username etc.)

DEP: 4

**2.4 Functional Requirement 4**

ID: 7

TITLE: Administrator add invoice

DESC: The administrator should be able to add an invoice

DEP: 4

**2.5 Functional Requirement 5**

ID: 8

TITLE: Administrator generate reports

DESC: The administrator should be able to generate monthly reports(income reports, overdue reports, payment reports etc.)

DEP: 4