Information system for taking care of plants ( PlantCare)

*Milos Mamic*

**Definition of the problem**

Let’s say you are a plant lover who takes care of his/her plant’s and from time to time you come to a certain situation that you need to leave you’re place of residence for short or long periods of time for work related reasons or the motive is something else. And all of your acquaintance’s are wrapped up with their jobs or out of town. Now you are in a sticky situation, aren't you?

You need someone that can take care of your plants for the extended period, and you do not know who to contact. In that case we got you covered. Plant care is here to take care of your plants for you. From professional people who love taking care of the plants to services that are cheaper and more affordable to the vast majority of people who do not have specific types of plants that require a lot of taking of for. Of course, the company ensures the customer that the people who are hired and are going to enter your humble home are virtuous and the company is eligible for the plant careers.

The plant careers would be interviewed by the company and signed with a contract that prohibits any type of disrespect to the customer. The company would be international and depending on the demand of the market it would spread out to all parts of the world. The service would be open to all types of customers that need it. Further down the road the company would offer basic courses for taking care of plant’s that are not so demanding to be taken care of so the number of employees would suffice the market at the given time .

**Functional requirements :**

The system should enable the following functionalities:

1. The system must allow the user to create their own account through a quick sign up / registration form with an e-mail address. An account is needed to experience all the features of the system.
2. The system should allow the users to be able to pay for a certain package and the payment would be connected with a bank account of the company(paying online directly)
3. The user could contact the company directly by email (via the site) to ask for any further help or questions about certain registered employees that are registered
4. There should be a rating system for the users to rate the employees that they encountered
5. The web site should be available in multiple languages for vas majority of the citizens of who which not all know English

**Non - Functional requirements :**

1. Since the system is going to be free, anyone could easily have access to it and create an account at any time.
2. The system should support two different types of users: administrators and users. The administrators can easily remove the users that are disrespectful to the careers.
3. The system must ensure 24/7 operations with an uptime of at least 95%.
4. The system should include documentation regarding the usage of the website.
5. The system design should be straightforward so that ordering is fast and easy.
6. The system should be developed and deployed within 3 months of systems designs being finalized.

**Feasibility study :**

The development of the system is economically viable since it does not require much expenses for its creation and sustainment. The only expenses come from the area of server deployment since it is planned for the website to work 24/7.

Regarding user data management, the system would require only the name and e-mail of the users. No other personal data would be collected.

From a technological standpoint, the most technically demanding part of this system would be the storage and management. A stable database is required for successfully storing the info and all its related ratings and comments(which would be sent via email). Other functionalities that are likely to be required are not considered as a problem.

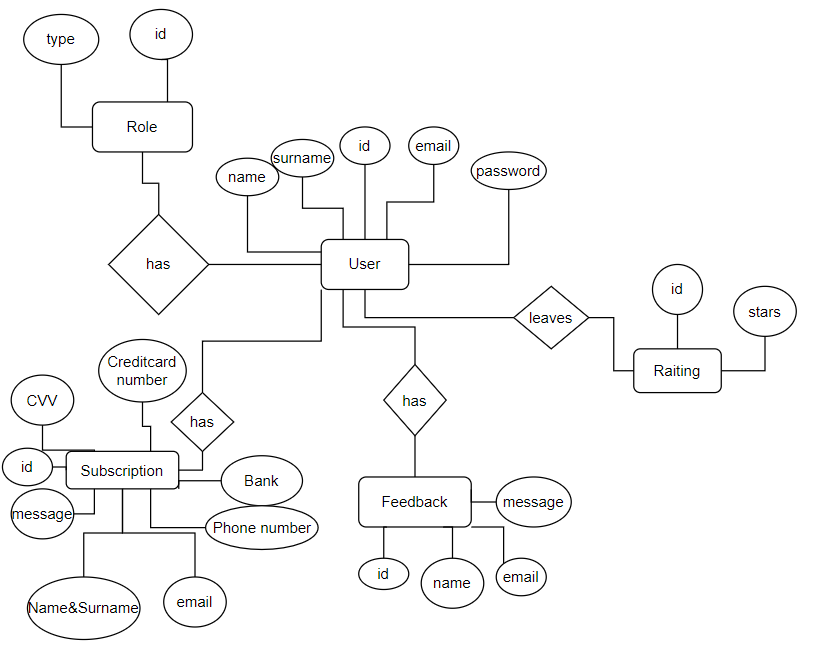
From the legal point of view there would be a contract which says that if the worker has caused any type of disruption upon the customer’s private property they are eligible and therefore they are required to financially take care of the costs.

These kinds of services can be found on craigs list or some other sources (e.g. friend of a friend) but there is no guarantee that they are not going to do the job property (professionally) without any doubt and or damage to the customer.

The users of the proposed information system would have a lot to benefit from. Everyone has the right to request their own services and since the usage of the website is completely free, they can save time for finding someone to do the job .

**Logical Design**

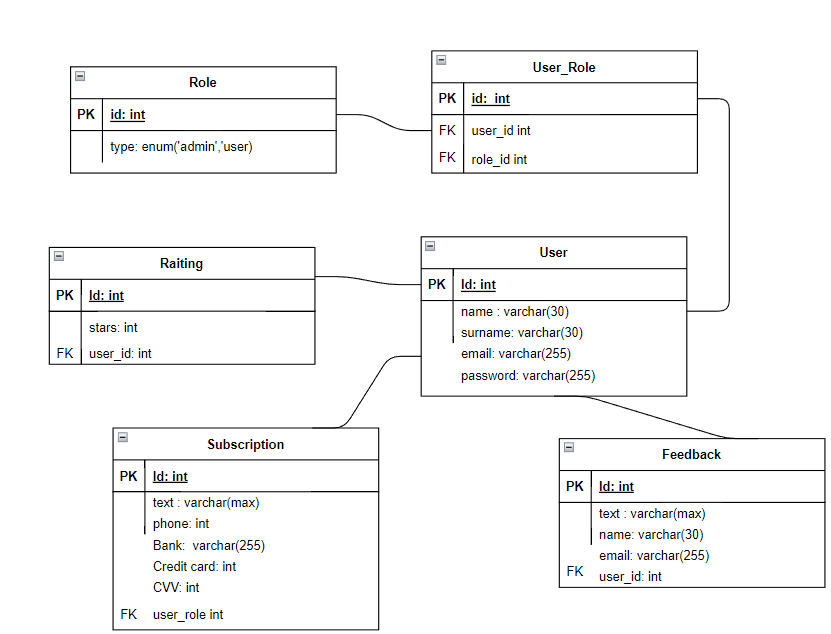
Entity Relational Diagram



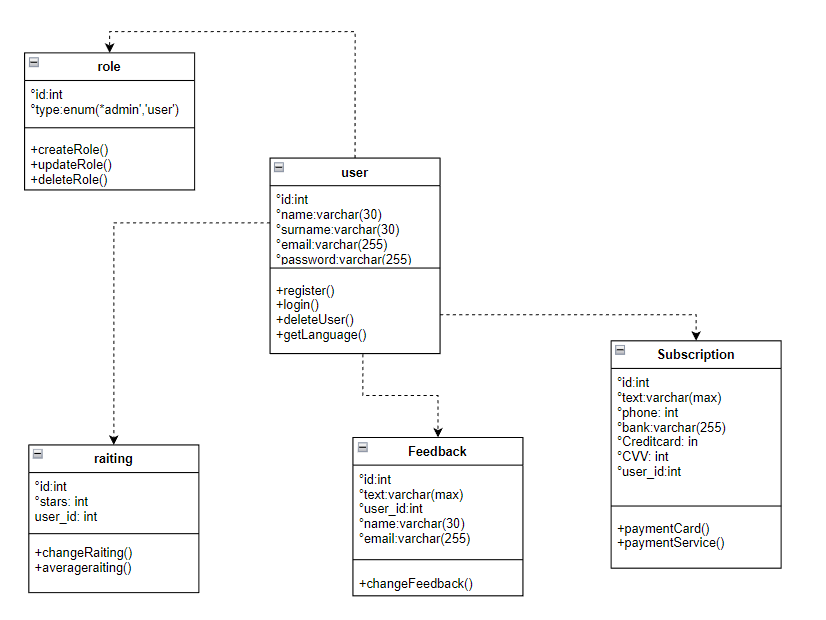
Matrix user role/function

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Functions | | User | | Administrator | | Moderator |
| Modify roles | | No | | Yes | | No |
| Delete account | | No | | Yes | | Yes |
| Leave rating | | Yes | | No | | No |
| Leave comment | | Yes | | No | | No |
| Change languages | | Yes | | No | | No |
| Login | | Yes | | Yes | | Yes |
| Sign in | | Yes | | No | | No |
| Entity | Description | | Attribute | | Type | | | Description of attribute |
| User | User of the system | | id | | int | | | Identity of the user |
|  |  | | name | | varchar (30) | | | First name of the user |
|  |  | | surname | | varchar (30) | | | Surname of the user |
|  |  | | email | | varchar (255) | | | Email of the user |
|  |  | | password | | varchar (255) | | | Password of the user |
| Role | The role of the ones using the system | | id | | int | | | Identity of role |
|  |  | | type | | enum(‘admin’,’user’) | | | Type of user |
| Subscription | Payment for the service | | id | | int | | | Identity of the payment |
|  |  | | text | | varchar (max) | | | Comment |
|  |  | | Phone | | int | | | Phone number of the user |
|  |  | | Bank | | varchar (255) | | | Name of the Bank |
|  |  | | Credit card | | int | | | Credit card of the user |
|  |  | | CVV | | int | | | CVV of the credit card |
| Feedback | Feedback from the user | | id | | int | | | Identity of comment |
|  |  | | text | | varchar (max) | | | Comment |
|  |  | | name | | varchar (30) | | | Name of the user |
|  |  | | email | | varchar (255) | | | Email of the user |
| Rating | Rating of the employee | | id | | int | | | Identity of rating |
|  |  | | stars | | int | | | Rating from 1 to 5 |

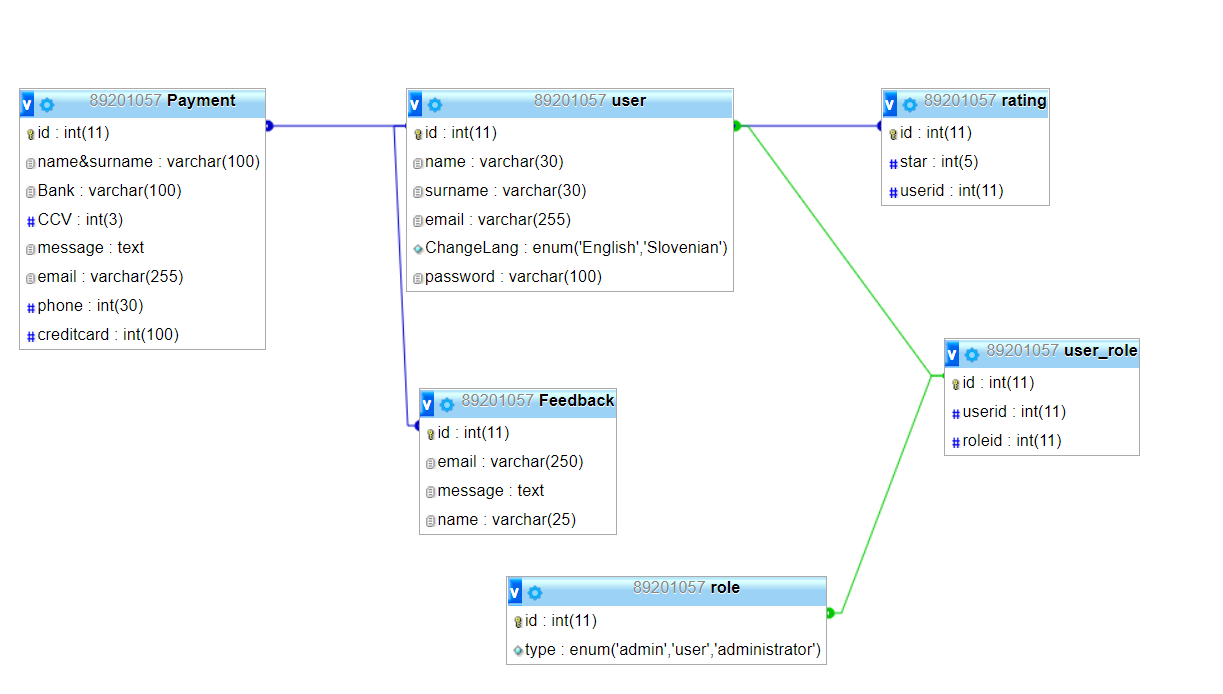
Relational model:



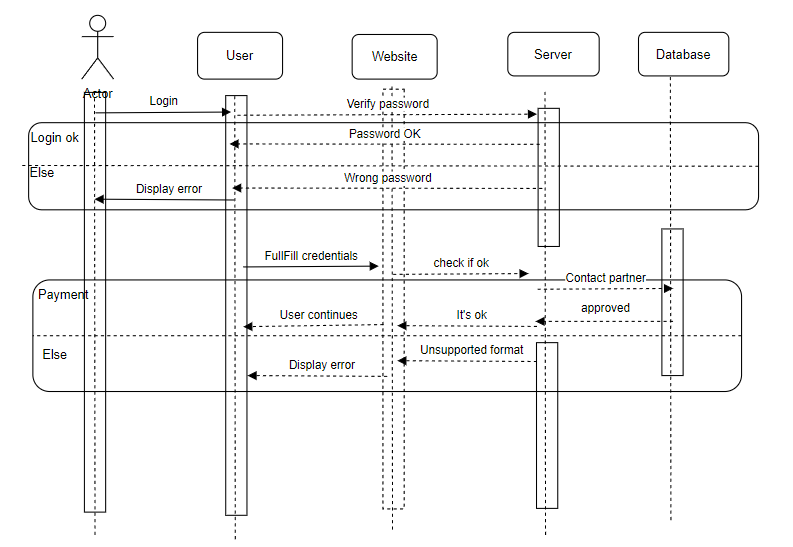
UML CLASS DIAGRAM:



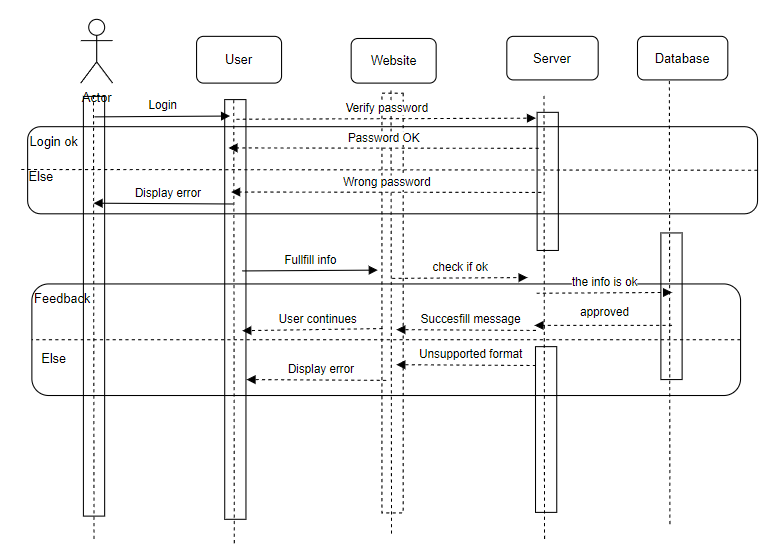
Physical Database Model



UML Use Sequence Diagrams



UML Sequence Diagram showing the process of payment



UML Sequence Diagram showing the process of giving feedback