

# Information system for reporting public nuisances

## Definition of the problem

People have problems with behavior of some individuals in the society who create public nuisances. There are many forms of public nuisances, let's focus for now on the most primitive ones, some local ones, that affect all of us. It can be, for example, some parking violation, which makes it difficult for pedestrians to walk, another example can be theft or some act of vandalism. Reasons for such activities are usually bad upbringing, poverty, mental disorders and many others.

None of the activities mentioned above are positive to the society, that's why they should be taken care of. As we can see now, we have two sides, one of which is people who do not appreciate such "violations" and suffer from them, and the other one is people who make such "mess".

The general problem, no matter how it sounds, is human behavior. Some of people bring negative consequences to the society, which society obviously does not appreciate, and wants to get rid of, which is, of course, very likely to be impossible, but what is possible is to reduce such troubles. The problem arises in ignoring such activities, which means that they will not end or even reduce. Instead, they can become even more frequent. In these cases, usual people are feeling the consequences as such problems do not bring anything good to them, even to the ones who do such things.

# Non-functional requirements

1. The system should be packaged as a web application.
2. The system must be able to process up to 26 000 report publishings per minute (approx. 26 000 residents live in Koper).
3. The system should provide a publication of each report in max 2 minutes.
4. The system should be developed and deployed within 6 months of system designs being finalized.
5. The system manages personal information of a sensitive nature and therefore must ensure safe communication between users and the server. Data exchange must always be encrypted.
6. The data needs to be stored in a secure way for a minimum of 5 years. The data should be stored on a minimum of two data carriers at all times. External backups should be made on weekly bases.
7. The system must ensure 24/7 operation with an uptime of at least 95%.
8. The system should support three different types of users: residents, authorities and moderators. Residents are the users who publish the violation report. Authorities answer these reports.
9. All users should be able to access the system from an arbitrary location using a personal computer, smartphone or some other device connected to the internet.

10. The system should include documentation for maintenance of the system for particular type of user.

# Functional requirements

The system should enable the following functionalities:

1. The system should allow a resident to publish a violation report for which the user can determine the name of the violation, the kind of the violation, the description of the violation and the proof (video/photo) of the violation. Any successful publishing is stored in the system.
2. The kind of violation mentioned above is chosen among the already existing particular kinds of the violations. The resident simply selects one of them.
3. The system allows residents to edit already published violation report. To do this, user first makes changes to the report, and then these changes are sent to the moderation to check them. Moderator decides whether changes are relevant to the main topic of the report or not (moderator either accepts or declines changes).
4. As mentioned above, the system allows a resident to give proofs. Proof of the violation is uploaded as files (photo, video, etc.). Up to 100 MB all together can be attached per violation report.
5. The system can be used by registered users only. You can only register as a resident type of user.
6. The system should allow the resident type of user to rate other violation reports, meaning that it is "promoted", so that it is on the top, and all the others violation reports

that are not that popular are on the bottom (sorting can be changed, of course).

7. One particular resident can rate some particular violation report only once. This is done to avoid multiple number of supports from one user.
8. The system should allow the authorities type of user to agree or not to agree with some particular violation report, depending on how the authorities user decides about particular violation. The decision "Agreed" or "Not agreed" is given to each report as time passes as authorities decide about it with a comment about it.
9. The system should allow all the types of users to browse through any prior violation reports as well as see all the decisions and comments for each violation report that authorities have made.
10. The user has an ability to give an appeal to the particular authorities' decision. The user should fill in text field where he explains why he does not agree with the already made decision.
11. The authorities have an ability to answer the appeal. They can change the already made decision to a new one, or not to change anything. They need to comment the new decision made.

# Feasibility study

One possible solution of the problem is to develop a web application that will enable publishing violation reports. Such reports should include the name of the violation, the kind of the violation, the description (text) of it and the photo/video files (proofs) of it. The violation report would be initially sorted from the most popular to the least popular (which can be changed). The popularity is dependent on the amount of the supports (rating) that each violation report has, which is, in its turn, dependent on how residents rate other reports. The authorities would see those reports and decide what to do with each violation report, whether to take actions or not. Residents would see their decision and the comment about it. The solution needs to satisfy all of the functional and non-functional requirements.

From a technical point of view, the project is quite possible to implement. The general appearance of the website can be implemented via CSS, in particular by using Bootstrap library. Functionalities that our website should have (e.g. violation report submission, authorities response) are not considered a problem as they can be handled by using server side scripting by using scripting programming language (e.g. PHP or JS). This will require a database and a web server, which are not that much expensive resources, at least if the amount of traffic is not too high.

The review of legislation revealed there are certain restrictions in terms of report publishings as it falls into personal information domain. The following limitations apply: (i) all data needs to be transmitted via encrypted link; (ii) data must not leave the EU or can do so if external institution has obliged to the GDPR rules; and (iii) data must be appropriately anonymized. After reviewing the above-mentioned technologies that can be used in our project from the previous paragraph, we conclude it is possible to implement the system in a way not to violate any of the above mentioned restrictions.

The proposed information system can be implemented on open source software. This allows us to conclude that a new information system does not require any additional costs regarding software or hardware part. In addition, the project makes productivity of the users that use it higher highlighting the economic viability of the project. In the end, the project is organizationally acceptable as it does not introduce any changes in the way work is being carried out at this point in time. However, the question of social acceptance is still open as the system can be abused by fake violation report, which is hard to prevent, as well as the proofs of violations can be faked.



## Matrix User role / functions

Functions	Resident	Authorities	Moderator
Publishing the violation reports	Yes	No	No
Changing the report and asking for accepting the changes	Yes	No	No
Accepting (or not) of changes made to the particular violation report	No	No	Yes
Voting for particular violation report	Yes	No	No
To give a decision to a particular violation report	No	Yes	No
To appeal the authorities' decision to the particular violation report	Yes	No	No
To answer the appeal (to give a new decision or to leave the same decision with a comment on it)	No	Yes	No

# Data dictionary

Table 1: Data dictionary

Entity	Description	Attribute	Type	Description of attribute
User	The human who uses the system	userID	int	Unique identifier of the user
		name	varchar(63)	The name of the user
		surname	varchar(63)	The surname of the user
Role	The particular role of the user	roleID	int	Unique identifier of the role
		type	enum	User type
Violation report	The violation report that resident has published	reportID	int	The unique ID of the report
		name	varchar(127)	The name of the particular report
		kind	varchar(127)	The kind (type) of it.
		text	text(max )	The description of it
		overallRating	int	Rating of the report
Rating	The rating of particular report	ratingID	int	The ID of particular rate made
		ratingValue	int	One particular rate itself (from one user)
Files	The files that were uploaded by some user for some violation report	fileID	int	The unique ID of each file uploaded
		rootDirectory	varchar(max)	Column to store file location
		fileName	varchar(max)	Column to store the name of the file
		fileAttribute	varchar(150)	Column to store file attribute

		fileCreatedDate	datetime	Column to store file creation time
		fileSize	numeric(10,5)	column to store the Size of the file
		fileStreamCol	varbinary(max)	Column to store the content of the file in the binary format
Answer	Authoritie's answer to the report	answerID	int	The ID of the answer
		decision	bool	"Agreed" or "Not agreed" decision
		comment	text(max)	The comment on the made decision
		created	timestamp	Time of answer created
Appeal	Report that has been asked for an appeal by some resident	appealID	int	The ID of the appeal
		text	text(max)	The description of the reason why resident thinks an appeal has to be approved
		created	timestamp	Time of appeal created
Not yet accepted Violation Report	Report that has been sent for editing, which means it should be accepted by the moderator	editID	int	ID of editing report
		name	varchar(127)	Name of the editing report
		kind	varchar(127)	Kind of editing report
		text	text(max)	New text of the report

*(Bold text in the table represents primary key in each entity)*

# ER Diagram

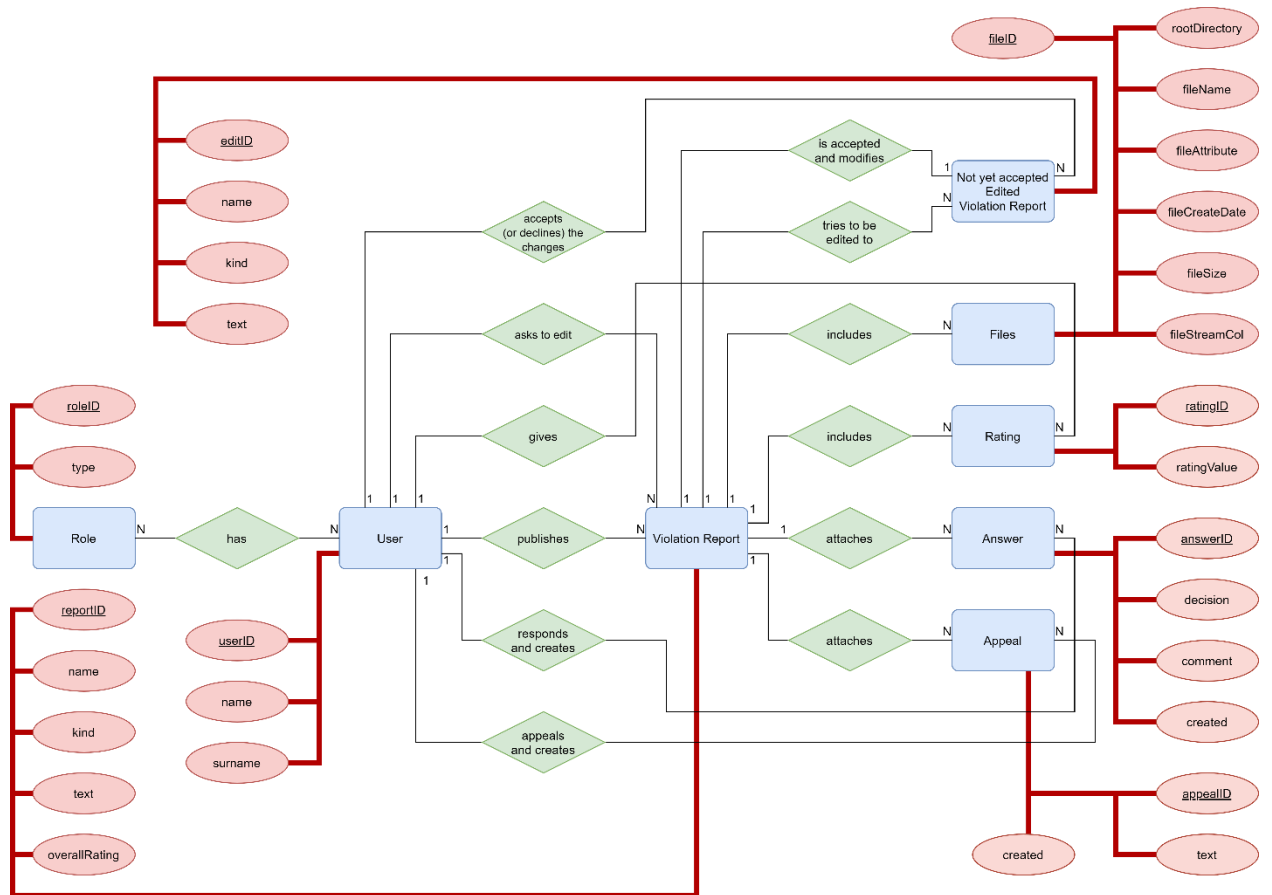


Figure 1: ER Diagram

# Relational model

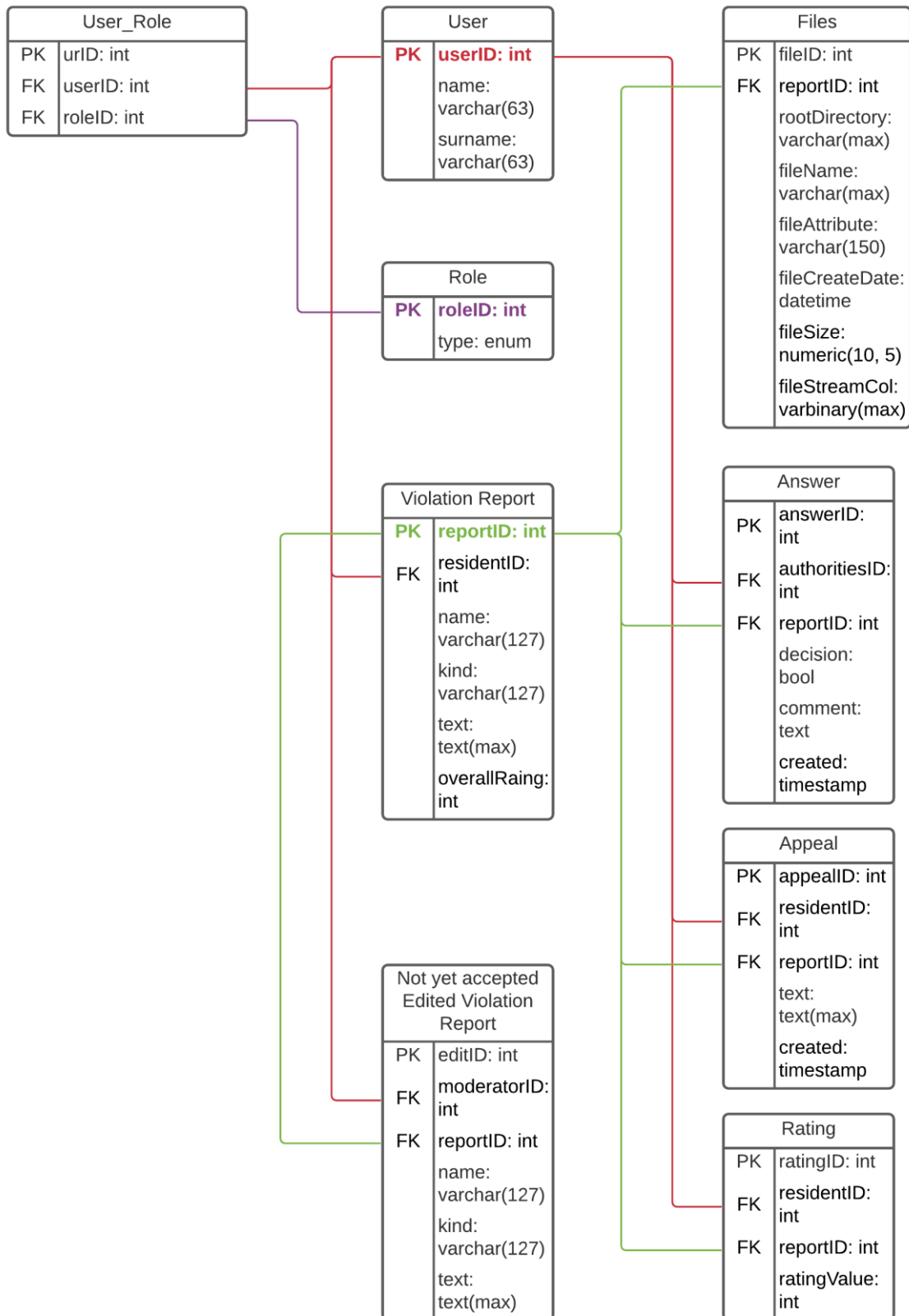


Figure 2: Relational model, where **red** lines represent **userID** key connections, **green** represents **reportID** and **purple** represents **roleID**

# UML Class Diagram

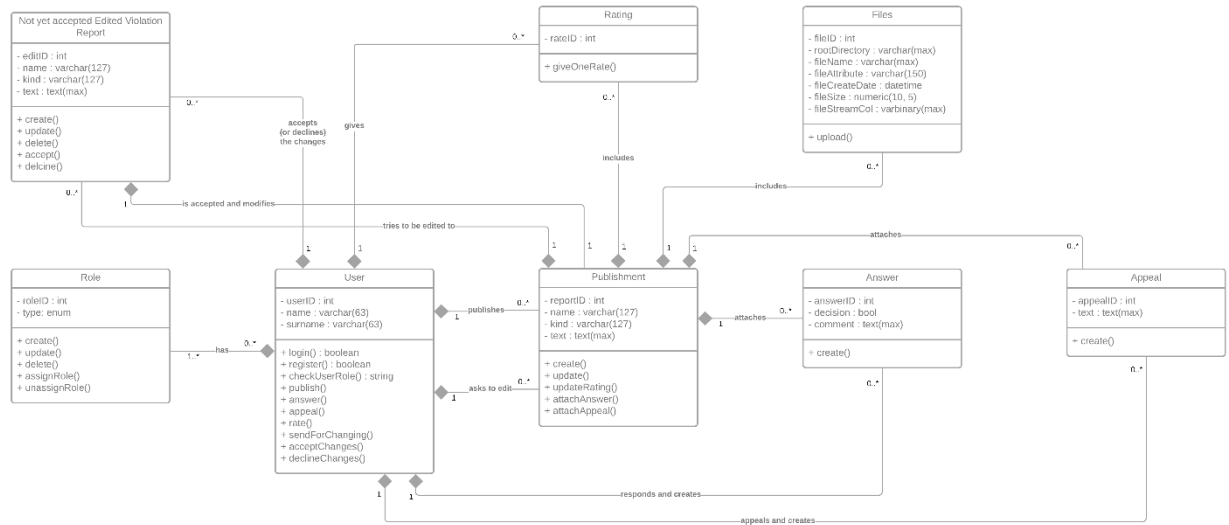


Figure 3: UML Class Diagram

# Physical Data Model

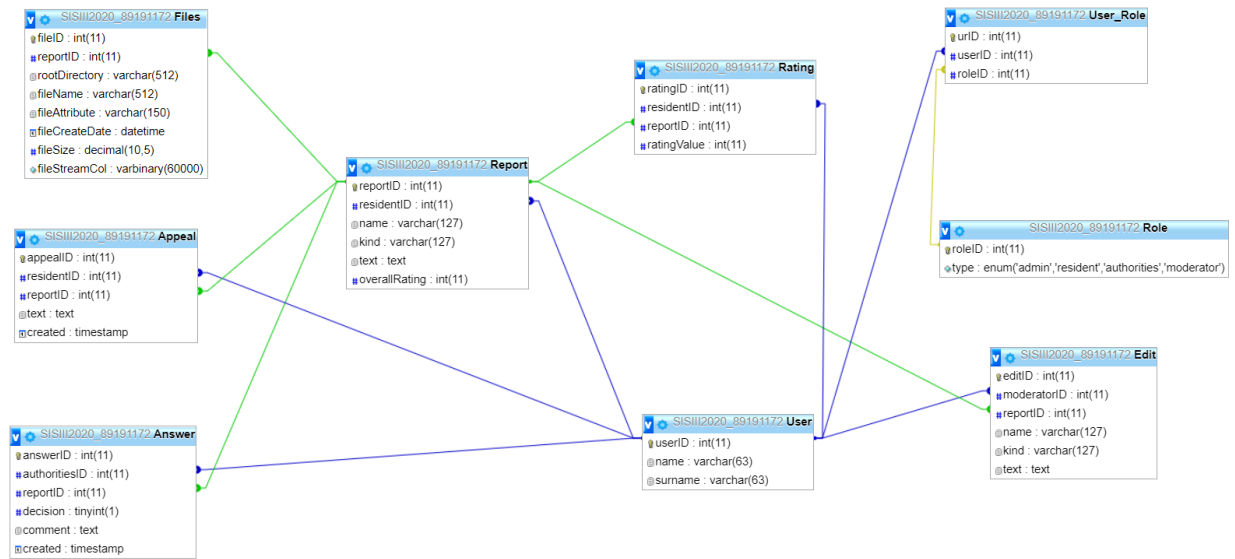


Figure 4: Physical Data Model

# UML Use Case Diagram

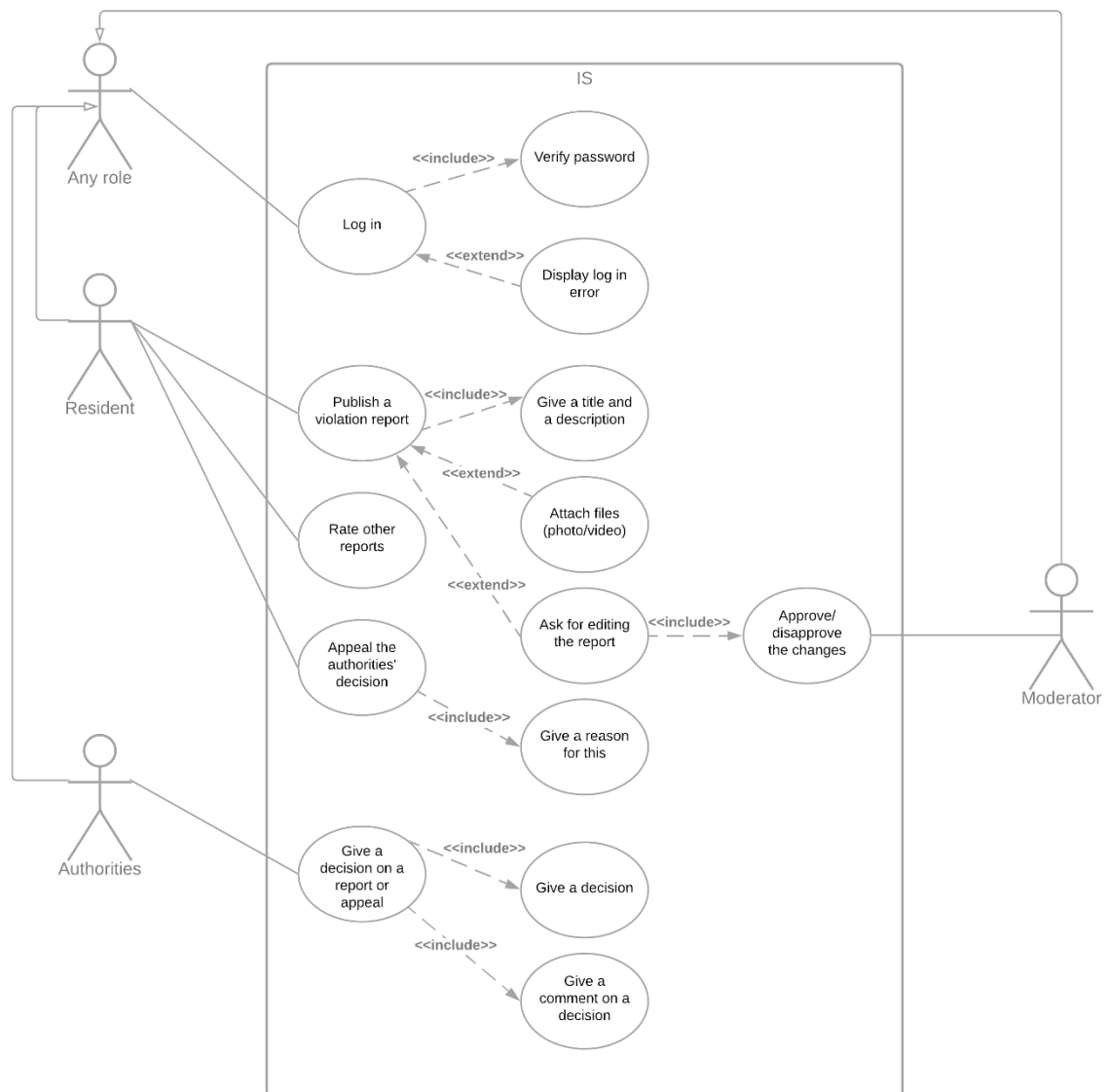


Figure 5: UML Use Case Diagram



# UML Sequence Diagram

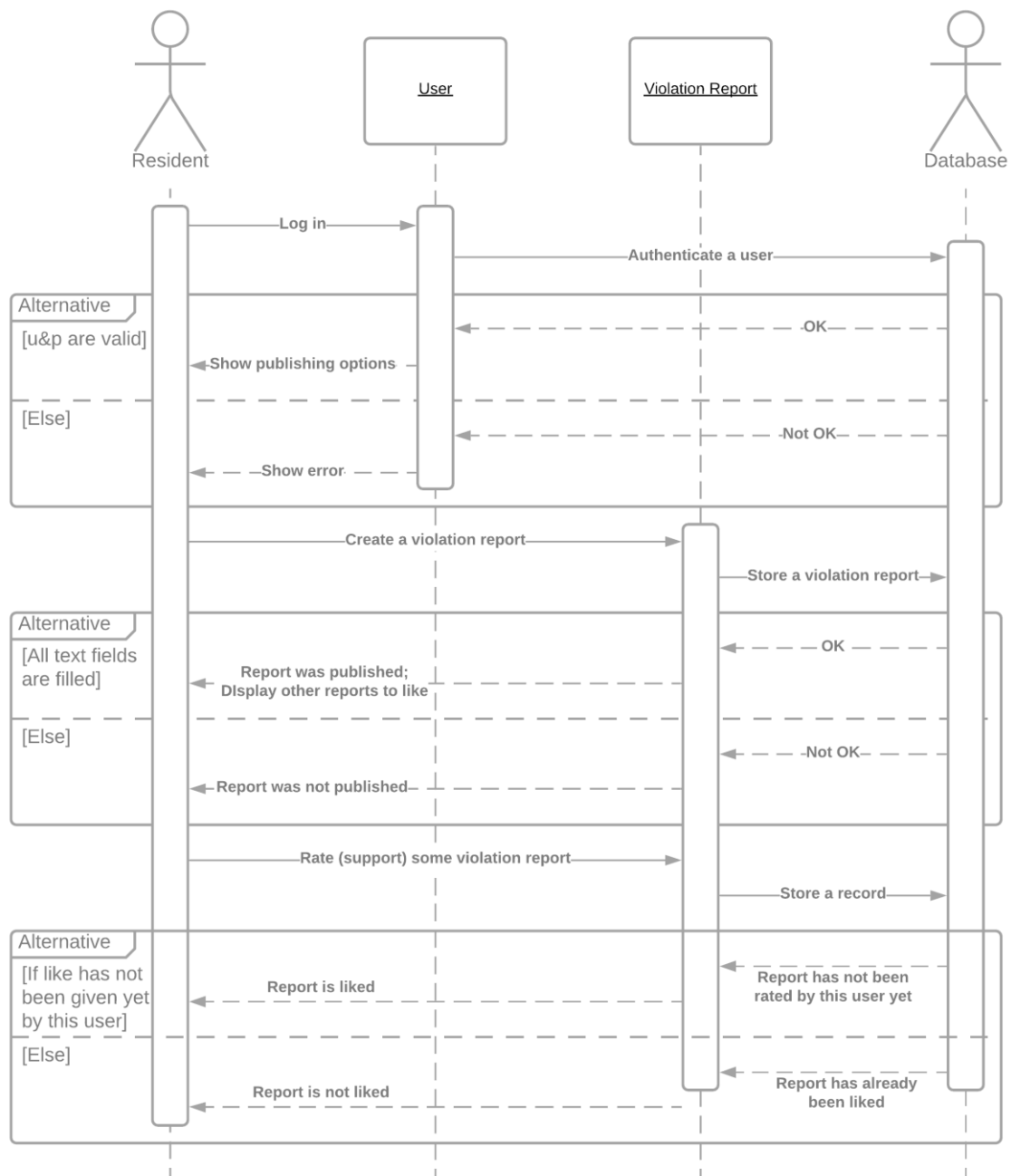


Figure 6: UML Sequence Diagram for a sequence of publishing a report and supporting (rating) some report

*See the next page for the second UML Sequence Diagram!*

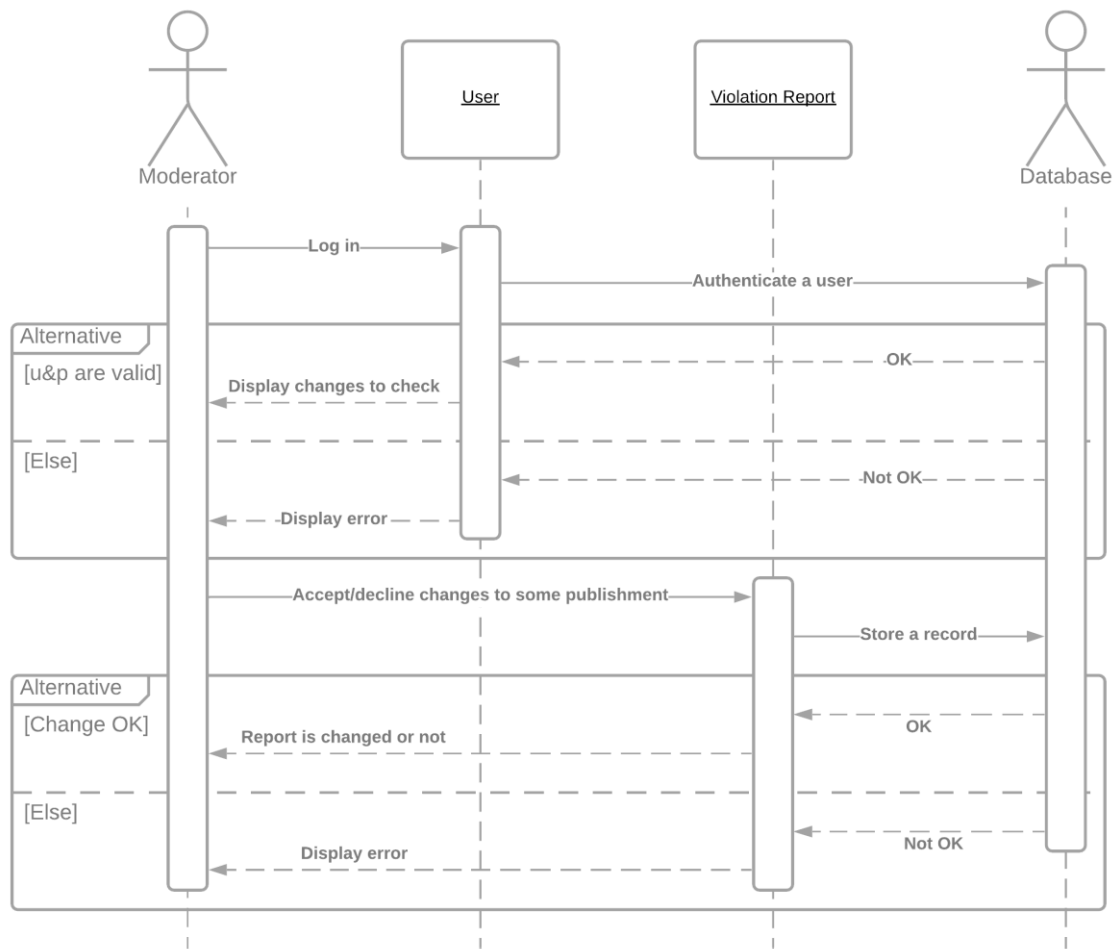


Figure 7: UML Sequence Diagram for a sequence of accepting/declining changes to some report