Software Development Methods
Project 2018/2019
Fix my town!

Miguel Goulão

September, 2018

1 Problem description

Many towns all over the world have recently adopted information systems that, via a web site or mobile applications, are making it easier to common citizens to report urban problems they want to see fixed in public places. Cities like Lisboa, Almada, or Boston, have applications for their citizens to report such problems. Usually, these applications allow citizens to report problems concerning trees, gardens, city equipment (cultural, sports, education, and social), roads, street signs, housing, urban hygiene, public lights, sidewalks, accessibility, sanitation, public safety, and noise, to name a few. The system typically offers the common citizen an interface the citizen can use to report a problem, but also listings with the history of reported problems and how they were followed up. A citizen may also search for complaints in a given area, and of a given kind (say, rats sightings near a kindergarten, or garbage near around the area of a famous night club). Finally, as a way of advertising the work the city hall is making to improve the quality of life in this town, the system may also provide a dashboard for this service, presenting, for example, a chart with the current status of the reported problems, as visible to the common citizen (under analysis, assigned for resolution, with an ongoing execution of a resolution, or solved). The system should include at least a web and a mobile interface for all relevant stakeholders.

When a problem is submitted, a trouble ticket is issued. The city hall analyses the problem description and decides on its validity. If the problem is considered a valid problem, it is then made available to the candidate fixers, so that the city hall can choose who will fix the problem and assign the problem to the selected fixer. All this is hidden under the umbrella of a problem "under analysis", as far as the citizen can see, but the city hall must have a more detailed view of this workflow. Once the city hall decides who will fix the problem, the problem is "assigned for resolution". When the fixer starts working on that particular problem, it changes the project status to "with an ongoing resolution". When the problem is fixed, the fixer has to notify the city hall of this resolution, so that the city hall, if necessary, verifies that the resolution was satisfactory. The city hall then moves the problem to the solved status. The problem reporters should receive notifications about the current status of their requests (naturally, only with the information visible to them, but not information reserved to the city hall and their partners).

The city hall uses the requests obtained via this application to monitor and allocate resources to solve those problems. This should include detailed dashboards for monitoring all the requests, using adequate filters, functionalities for supporting the workflow of those requests, and a back office to manage the whole system. In particular, the city hall has to be able to manage the problem hierarchy and, if necessary, add new problem categories and sub-categories. The city hall has also to be able to add geo-referenced information to the city map (e.g. updating the

city map, adding a school in a particular neighborhood, adding restrictions to the way traffic flows in the city streets, and so on).

There are many problem fixers. Some of the solutions are directly provided by the city hall services, while others may require the intervention of external entities, such as a fire department, a parks and recreation company, or a constructing company, among many others. The city hall has to register its own services. The partner organizations will need to register themselves in the system and select the problem categories in which they can help. This information needs to be validated by the city hall. Once validated, the entities will receive alerts of new issues in their area of intervention. They can only apply to solve problems within their areas of intervention. The city hall is responsible for assigning problem resolution to one of the candidate services, be those provided by companies, other external organizations (e.g. fire department), or city hall services. All of these receive problem alerts, so that they are aware of opportunities to help improving the city.

Finally, the system should also support some kind of feedback, so that citizens can rate their satisfaction with the problem solution. This will allow the city hall to have, within its dashboard, the ability to assess the level of satisfaction with the problem resolutions in different areas, and even filtered by problem fixer, allowing the city hall to know which are the best (and worst) fixers.

Disclaimer: This text does not describe exhaustively the problem. It is deliberately open for each group to interpret and complete with its vision of how such a system should work. Be creative. Your model must comply with the information in this document, but you can and should expand on it. Think of these as "minimal requirements". If you think you found any ambiguity or undefined issue in this description, good. This description is supposed to be like that. Call for project proposals are like that, in real life. Feel free to discuss ambiguities and open issues with the lab teacher.

2 Intermediate report - 1st phase (Due 2018/10/18)

The report shall comply with the following guidelines:

1. Front page with the name of the course, title of the work, name and number of the team members and date. Besides, it should explicitly state the lab where the group is enrolled and name of the teacher that is lecturing the lab.

2. A chapter with a list of assumptions found relevant to explain options made (shouldn't go over 1500 words). These assumptions should cover issues that the group may have found as open, or even ambiguous, in the high-level problem description provided in this document.

3. A chapter with the results of the analysis phase organised by sections:

   (a) 1 complete Use Case Diagram

   (b) 3 complete Use Case specifications, including 2 alternative scenarios per use case. **Hint:** make sure you choose interesting use cases to model. Trivial, or too common use cases (e.g. logging into the system) are **NOT** interesting for this report, because you can normally reuse them from one project to the next. Choose use cases which are specific to **THIS** project.

3 Final report - 2nd phase (Due: 7/12/2018)

The report shall comply with the following guidelines:

1. Front page with the name of the course, title of the work, name and number of the team members and date. Besides, it should explicitly state the lab where the group is enrolled and name of the teacher that is lecturing the lab.

2. A chapter with a list of assumptions found relevant to explain options made (should not go over 1500 words). These assumptions should cover issues that the group may have found as open, or even ambiguous, in the high-level problem description provided in this document.

3. A chapter with the results of the analysis phase organised by sections:

   (a) 1 complete Use Case Diagram

   (b) 3 complete Use Case specifications, including 2 alternative scenarios per use case. **Hint:** make sure you choose interesting use cases to model. Trivial, or too common use cases (e.g. logging into the system) are **NOT** interesting for this report, because you can normally reuse them from one project to the next. Choose use cases which are specific to **THIS** project. The Use Cases should be agreed with the Lab teacher, to be sure that the selected use cases have enough relevance and complexity to be accepted.

   (c) 1 Activity Diagram per Use Case must be the same as the ones selected in b.

   (d) 1 Sequence Diagram per Use Case detailed in b).

   (e) 1 Class Diagram (Domain Concepts, not the architecture design with control, boundary classes etc). This should be a fragment diagram that covers completely the classes and required relations used in the Use Cases selected in b).

   (f) OCL about the Class Diagram e) with reasonable complexity properly discussed with the teacher in the lab: 3 invariants, 3 Pre-conditions; 3 post-conditions.

   (g) Statechart Diagrams per control Class.

   (h) 1 Package Diagram (a global one and others if necessary).

4. Chapter with the results of the design phase, organized in sections. Suppose the implementation will be using Java with Oracle Databases and a three level architecture.

   (a) Class Diagram clearly differentiating Entity, Control and interface Classes

   (b) 1 Package Diagram (architecture)

   (c) 1 Component Diagram

   (d) 1 Deployment Diagram

   (e) 1 Relational Database Schema

5. Chapter with conclusions (limit 500 words). This can contain self assessment statements that, in retrospective, can include main limitations and difficulties felt during the project along the semester.

6. Bibliography. Book/white papers/ technical reports/other references and links of websites used to gather information either related to the Domain of the problem or to the domain of the solution.

4 Project submission

1. The date for submitting in moodle the written report in pdf for the first phase is: 2018/10/18 at 23h55 (hard deadline). For every 3 hours past this deadline, you will receive a 1 point penalty in the classification (*i.e.* 8 points per day).

2. The date for submitting in moodle the written report in pdf for the second phase is: 2018/12/07. For every 3 hours past this deadline, you will receive a 1 point penalty in the classification (*i.e.* 8 points per day).

3. We will schedule oral discussions for defending this work is scheduled for the week of 2018/12/10 to 2018/15. **The presence of all the three team members is mandatory**. All team members must be knowledgeable of the whole content of the written report to be discussed. The discussions will happen after the delivery in the second phase. For that you should go to the secretary of the department after submitting the project and block your slot in the agenda in paper that will provided to you there.

5 Deliverable format

In both phases, your project is to be submitted on Moodle. In order to create it, you will need to use appropriate model and text editors. No hand-written reports or diagrams within those reports will be considered.

1. In the first phase, please submit your written report **in *pdf* format (no other formats are acceptable)**. The *pdf* file must be named as:

   [Lab number]-[student 1]-[student 2]-[student 3].pdf

   For example, for a group enrolled in the lab P2, we would have:

   P2-34234-65444-83221.pdf

2. In the second phase, please submit a *zip* file named:

   [Lab number]-[student 1]-[student 2]-[student 3].zip

   For example, the same group would be identified as

   P2-34234-65444-83221.zip

   The zip file should contain:

   (a) the written report **in *pdf* (no other formats are acceptable)**
   (b) the *.use* file (and *.soil* file with the instances used for testing your models with OCL)
   (c) any other document found relevant

6 Useful links

Here are a few links to sites/apps with relatively similar purposes compared to those in this project.

1. Lisboa: `https://naminharualx.cm-lisboa.pt/`
2. Almada: `http://www.m-almada.pt/almadamaisperto`
3. Boston: `https://mayors24.cityofboston.gov/`