Project Specification

02160 – Agile Object-Oriented Software Development

v. 2019-3-12

1 General information

You have been asked to develop software to run hospital information management. The system has to provide the mandatory functionalities listed below. Moreover, a set of optional functionalities has to be implemented (at least two). They can increase the value of the system but they cannot compensate mandatory ones.

You can either choose from the list of optional functionalities, or suggest your own ones. The optional feature set to be implemented has to be negotiated with the teachers upfront and will depend on the complexity of the chosen features and the group size.

2 Functionalities

2.1 Mandatory functionalities

ID	Title	Description
M1	Patient registration	The system should be able to register patients to the hospital. Some information should be entered by a registration clerk (e.g., name, surname, birth-date, home address, phone number, tribe, alive/dead), while some information should be automatically created by the system (e.g., incremental patient number). The system should allow to sign-up patients and change patient information (e.g., their address). Moreover, the system should allow finding registered patients based on simple criteria (e.g., name, surname, birth date).

ID	Title	Description
M2	Staff management	The system should be able to register staff working at the hospital. Basic information should be entered (e.g., name, surname, job role: doctor, nurse, clerk, ICT officer), while some other should be automatically generated by the system (e.g., serial number, unique name/surname-based email). The system should allow registering staff and changing their information (e.g., when they change to another ward or job role). Moreover, the system should allow finding staff members based on simple criteria (e.g., name, surname, email).
M3	Health facility management	The system should be able to represent the different departments available in hospitals. Departments might have a certain number of beds available. The system should allow retrieving the number of beds currently in use. Health care staff should belong to a single department.
M4	Patient admission	When patient is registered upon arrival, they are admitted to a department at the health facility. A patient is received at various departments or units in the health facility such as at the emergency, outpatient or inpatient. Upon receiving a patient in an inpatient department nurses will allocate the patient to a bed whereas in outpatient departments, the patient is received when a doctor calls them. Patients can both move between departments and beds (within the department or across departments). Patients can also be discharged. In the case of moving between departments, patients are first discharged from their current department and admitted to the new one.
M5	User interface	It should be possible to interact with the system via a graphical user interface. Data correctness should be guaranteed through proper checks (e.g., birth-date has to be a date in format dd.mm.yyyy). Moreover, the user interface should cover finding of the entered data.

Note that for the mandatory functionalities, it is only necessary to model an instantaneous snapshot of the current situation at the health care facility.

2.2 Optional functionalities

ID	Title	Description
O1	Scheduling appointment	The appointment scheduling process allows health worker to make patient appointment with preferred date and time to obviate the necessity to sit in queues and wait for consultation. The patient contacts the Health facility by agreed means of communication; say by telephone, personal contact, email etc., and books an advance appointment with the choice doctor/department on a given date and given slot of time.
O2	User management	User management and role-based access: health care staff see their patients and patient admission data but not staff data, clerical staff only see data required for patient registration and admission (no health data), the ICT officer sees all.
O3	Patients waiting	The system should allow clerks, doctors and nurses to query the system to know which patients are admitted to outpatient but have not finished their treatments for the day.
O4	Persistency layer	The data (i.e., patients, doctors, admission data, registrations, etc.) should be stored in a persistency layer (e.g., a database, a XML/JSON file). This way it is possible to close the system and open it again with the status preserved.
O5	Participation lists	The system should allow creating participation lists for health facility departments, including all patients admitted to a certain department at the moment. The generated list should be used by clerks to verify patients in a department and should provide (customizable) information. The list should be produced as ODT[1], PDF or another easily readable file format.
O6	Advanced query mechanism	The system should allow its users to query it based on "advanced properties" that involve investigating the interaction among several entities (e.g., queries involving the relationship between departments and patients). Example of such queries are *"departments with most patients admitted"* and *"departments with highest/lowest number of waiting patients"*.

3   Practicalities and assessment criteria

The project has to be done in teams of 4-6 students. The project has to be delivered together with the final presentation on the last week of the semester.

Each group should deliver:

- Report (up to 15 pages) containing

    - User stories
    - UML Class diagram
    - Description of software architecture and design (e.g., patterns used)
    - Project management details (including how the project was conducted and how the software quality was ensured; with reflections on how agile practices were used)
    - Work distribution among project members
    - User manual
    - URL to a screencast of the running software explaining all implemented functionalities

- Software on Git repository including the following

    - Documented Java source code
    - Cucumber tests
    - Unit tests
    - Executable `jar` (uploaded as GitHub *release*)

Test coverage should be at or very close to 100%.

The project will be presented by each group in the last week of semester. Details regarding the presentation will follow. In addition, there will be two peer review sessions, on 27 March and 24 April. The deadline for group formation (done online) is 19 March so everyone will be able to start project work at next week's exercise session. It is recommended to start the project with a planning meeting where group's working methods and style are discussed, agreed on and written down.

The final grade will consider the implemented functionalities and the quality of the software as well as the quality of the report, and individual oral assessment.

---

[1]See `https://en.wikipedia.org/wiki/OpenDocument`.