

# Software Design

Midterm

November 21<sup>st</sup>, 2017



*During this Exam I will not undertake any illegal acts of accepting or providing any solutions to other students. I also state that my health condition is good and that I am capable for taking this Exam.*

---

*Student ID*

---

*Name and Surname*

---

*Hand-written signature*

1. (1 mark) Software process activities where customers and engineers define the software that is to be produced and the constraints on its operation is called \_\_\_\_\_.
2. (1 mark) Software requirements are classified as: \_\_\_\_\_, \_\_\_\_\_ and \_\_\_\_\_.
3. (1 mark) Name at least three ways of software requirements specification.  
\_\_\_\_\_  
\_\_\_\_\_
4. (1 mark) Use case scenario typically includes:
  - precondition description,
  - description of regular (standard) flow,
  - \_\_\_\_\_
  - information regarding parallel activities,
  - description of scenario result.
5. (1 mark) Which connection is used between actors and use cases?  
\_\_\_\_\_
6. (1 mark) Draw graphic symbol which represents object destruction in sequential diagrams.
7. (1 mark) Generalization is in use-case diagram a valid relation between \_\_\_\_\_.

8. (1 mark) Time limited period of 30 days or less, during which a complete and potentially shippable product increment is produced in agile Scrum method is called \_\_\_\_\_.
9. (1 mark) For which software engineering process model the reuse-oriented development is the key characteristic? \_\_\_\_\_
10. (1 mark) During class diagram modelling in object oriented software design, a part of visible behaviour of a class of objects is described using \_\_\_\_\_.
11. (1 mark) Liskov substitution principle states that a program must execute correctly if an object of \_\_\_\_\_ type is assigned to a variable of \_\_\_\_\_ type.
12. (1 mark) In software architecture design the system structuring process includes the definition of subsystems and \_\_\_\_\_ mechanisms.
13. (1 mark) In Git collaboration tool new data is retrieved from the remote repository into \_\_\_\_\_ repository using the command \_\_\_\_\_.

### **Problem solving – Smart City Parking System (SCPS)**

Smart city system requires an automated way of finding car parking space. The envisioned solution - **Smart City Parking System (SCPS)** is based on crowdsourcing and relies on data gathered from registered users via smartphone application.

Each user must register with email address when first using the smartphone application. During registration process the user may also define his/her vehicle characteristics if that vehicle has the ability to identify a free parking space during driving (see picture below). Each user can register up to three vehicles.

SCPS gathers information about free parking space positions from modern car drivers during the ride via smartphone application which connects with the vehicle for data gathering. This process is facilitated via universal car interface (a part of the vehicle) which collects the data from ultrasound parking sensors and/or integrated cameras and recognizes a free parking space.

The smartphone application is the data provider (DP) to SCPS for:

- Locating free parking space position – app receives data from the car sensors
- Notifying the SCPS about occupying or leaving parking space – app receives data from the car sensors
- Notifying the SCPS about occupying or leaving parking space – app receives data from smartphone sensors for vehicles which do not have their own sensors.

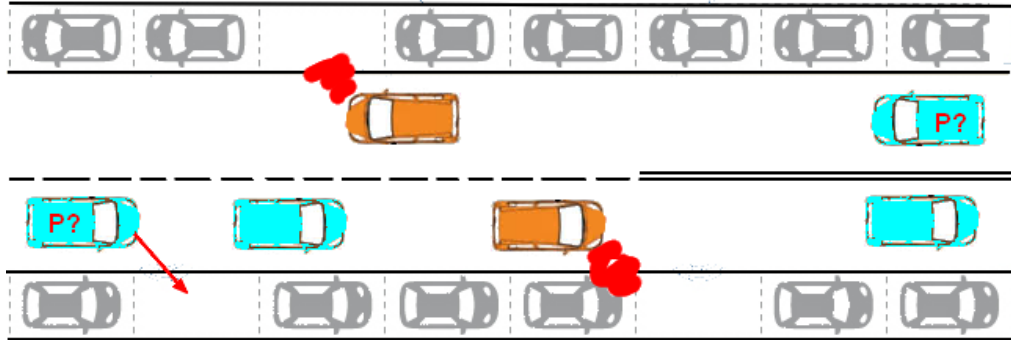
The received data is stored in the central database of SCPS.

The smartphone app is the data receiver (DR) from SCPS for:

- Available parking space positions.

Registered users can track available parking spaces using the smartphone application. In order to display available parking space the smartphone application must retrieve the data from the database and integrate it with Google Maps.

SCPS website displays a map of the parking spaces for an area and the information about current percent of parking space occupation. Each area can have multiple subareas, whereas each one has one or more parking spaces. Each parking space can be in one of the following states: *free*, *occupied*, *in process of being occupied*, *in process of being freed*. The type of each parking space is either *regular* or *for people with disabilities*. A parking space for people with disabilities can only be reserved by a user which has a special permission. During the map search of a subarea, users which have already registered via smartphone app can announce their intention to park in that subarea and specify time frame. They can also delete an announcement previously made. Each registered user can have up to three active parking announcements. Registered users can also change their registration data on the website.



**NOTE:** When modelling diagrams along with general requirement also observe additional requirements explicitly stated in each task.

#### 14. (4 marks) Use case diagram

Model the use case diagram for Smart City Parking System. Do not include website functionalities in the diagram!

#### 15. (4 marks) Sequential diagram

Model the process of occupying parking space and notifying SCPS about it.

When modelling the diagram take also the following requirements into concern:

- The process of occupying parking space starts when vehicle speed falls below 10 km/h.
- The smartphone app contacts the SCPS to find a free parking space nearest to the smartphone current GPS location. SCPS finds the locations of free parking spaces nearest to the location received from smartphone app by starting a new database search process.
- Upon receiving the answer from the SCPS, the smartphone app uses an internal algorithm to notify SCPS of successful parking space occupation. The SCPS stores this information in the central database.

#### 16. (4 marks) Class diagram

Model the SCPS website using UML class diagram. List class attributes and operations and indicate their visibility. Also indicate multiplicity for class relationships. Do not model the data types.