DIT184 Software Analysis and Design

Software Engineering Division, D&IT

June 1st 2018

Time:	08:30 - 12:30
Place:	Lindholmen
Teacher:	D. Stikkolorum, M. Chaudron
Max. score:	100
Grading scale:	G=55, VG=75
Exam. aids:	dictionary

The examination consists of 2 parts:

1. 2 theory (T1 and T2) questions related to the course topics, and

2. 5 modelling problems (P1 – P5) related to the case of a single system
    - Some problems build on each other; it's recommended to consider them in order.

Practicalities

Start each question/problem on a new sheet. (Sub-problems, may be put on the same sheet).
Only write on the front of each sheet.
Label each sheet with

- your *anonymous code (provided by the attendant).*
- the *problem number, i.e., T1,T2,P1, P2 ...*

Before handing in:

- sort your sheets in problem order, and
- enumerate sheets as 1,2,3,....,
- check that your anonymous code is written on each sheet.
- **write down/memorize your anonymous code** (to check anonymous exam results later)

Good Luck!

Assignment Nature Park Wildlife App

The Bainomugisha Nature Park (BNP) is one of the biggest nature parks in Africa: it covers several thousands of square kilometres with different types of ecosystems: desert, savannah, rainforest. The park contains a lot of wildlife – of various types: birds, butterflies, grazers (impala, giraffe, buffalo), monkeys, and predators (lion, leopard). The park contains a campsite and lodges catering for various categories of travellers (low-budget, families, luxury travellers).
The Bainomugisha Nature Park and reserve is very IT-savvy. They are the most advance nature park in Africa in terms of IT. They also care a lot about the well-being of their animals.

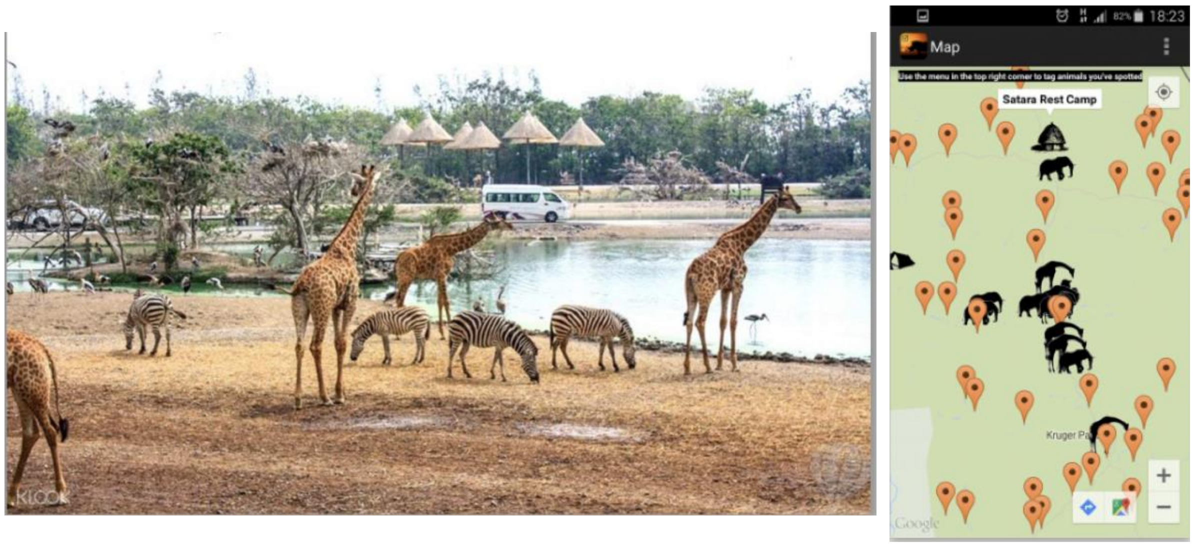

Map
Use the menu in the top right corner to tag animals you've spotted
Satara Rest Camp
Kruger Pa
Google

Figure 1 Left: wild park - Right: wild life app

The BNP is offering an app for the guides of the park and also to the visitors to the park. Upon entering the park, visitors create an account in the BNP system. In this account, they register the mobile phone numbers that they use and link this to the licence plate of the car in which they are driving around through the park.

The app offers the following functionality:

- **General Information:** the app provides general information about the park: based on the GPS-coordinates, the app can offer information about the particular geography and ecosystem of the area where the visitor is. Also, it provides background information on animals: what there typical habitat and behaviour is and how to recognize them. The maps also show information about camp sites, fuel stations and first-aid/medical stations.
- **Sharing spottings:** Live notification of animal spottings: when people spot an animal, they can enter this event (time & location (GPS) are registered by the smart phone). These spottings will be plotted (within reasonable time) on a

map. Also, visitors can subscribe to receiving notifications about particular types of animals.

- **Prevent crowding/flocking:** The BNP wants to prevent crowding/flocking – i.e. too many cars going to the same place. Hence the system shall stop forwarding and plotting of spotting-notifications when the number of safari-cars (set by some threshold) in the proximity of an animal is too high.
- **Checklist with image recognition:** The app includes a check-list of animals that can be found (see figure 2). This checklist is organized by animal-type. Visitors can check items manually or they can upload photos of the animal and the system will do image-recognition to determine which type of animal was spotted. For this image recognition, the system utilizes an enormous databases of several Terabytes of labelled photos.

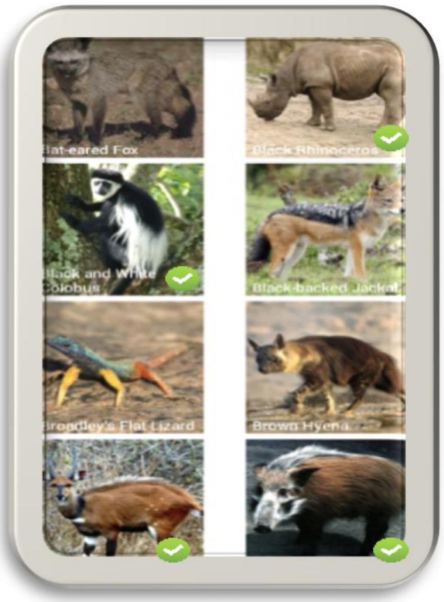


Figure 2 Example checklist of animals spotted

- **Control Centre:** There is one control room in the park. This room has one large display that shows the sightings of animals and the positions of all cars in the park. The control centre also collects data of all spottings of all animals over time so that this can be used for monitoring animal populations in the park.
- **Panic/Emergency-button:** When someone gets attacked by a lion, they can press the alarm-button on the app. Alternative uses are to trigger the emergency-button *after* an attack. The emergency signal is forwarded to all rangers in the park and to the medical staff that is located in the nearest staff-quarters. The officer in the control centre then looks at his screen and selects (based on proximity) which guides & medical officers should go to the place of the emergency.

System Requirements

For the development of the app some requirements are written down:

Application (system) requirements

- In the future, the system should also be able to support other kinds of animals than described in the case text. In this way the app could be used for other parks or when the collection of animals in the park grows.
- The application uses a modern graphical user interface.
- In general the software should be maintainable, extensible and changeable.

Technical requirements

- The program language for implementing the system is Java (object-oriented language)
- The system uses a couple of pre-defined software classes:
    - **GPSController**, is responsible for handling the signals from the physical GPS sensor.
    - **ImageRecognizer**, is capable of comparing images with images from an external database.
    - **Canvas**, the GUI module that is responsible for showing the graphical components of the application (such as buttons and text-messages).

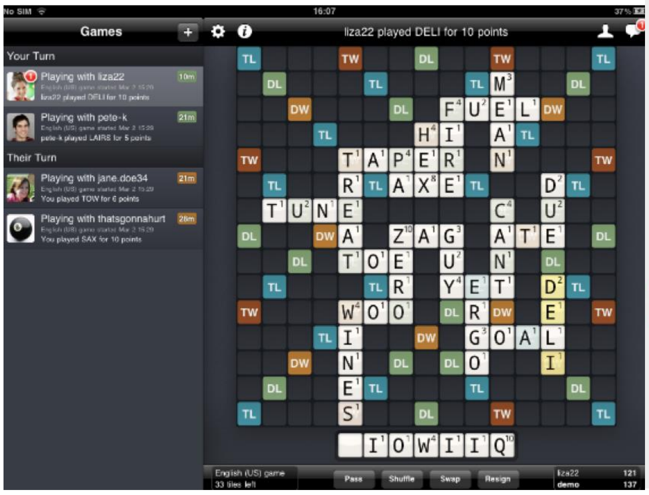
  The Canvas consists of several implementations of:
    - **GraphicalComponent**, a reusable class that is responsible for handling events for graphical components. **Button** and **MessageArea** are specialisations of this class.
    - There should be a class that has a role as central control unit

T1. Software Analysis and Design (15)

This course is called ‘Software Analysis and Design’. Based on the reading material of the course and the lecture slides:

a) What is the purpose of ‘Analysis’?
b) What is the result of performing an ‘Analysis’?
c) What is the purpose of ‘Design’?
d) How do analysis and design relate to each other?

T2. Recognize the commonality (15)



Wordfeud is a 2-player game that people can play via their phone. Both players get a random selection of letters that come from a (hidden) bag of letters. When players get these letters they are removed from the bag. In each turn, players can use these letters to place a word on a grid. The word needs to connect to an existing word. Placing a word gives a score. This score is determined by: 1) the values of printed on the letters (‘easy’ letters have a low value, and ‘difficult’ letters (like ‘X’ and ‘Y’) get high values, 2) some special tiles on the grid have special effects on the score: when a letter in a word covers a ‘DL’ or ‘TL’ tile on the grid, then this doubles, respectively triples the score of the word.

Crossword puzzle: there is a matrix of squares. A square can be black or white. While squares may contain a number. For each number there is a description of the word that needs to be written in the row or column that starts with this number. To solve a crossword puzzle, one must write words for all white squares – such that the horizontal and vertical words are compatible.

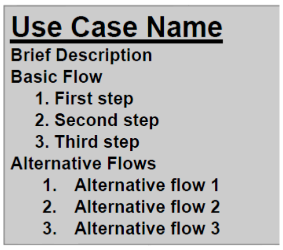
Question: suppose you were to do a domain analysis (having in the back of your mind to build a system that would implement these games as software), then which concept or relations would be common to both the wordfeud case and the crosswords case? Name 4 such concepts/relations and explain your choice.

The following questions relate to the Nature Park Wildlife App case.

P1. Use Case based requirements (15)

a) Make a use case diagram and include at least two sensible use cases in addition to the use case '*Check Spotted Animal*' (see 'Checklist with image recognition'). '*Share Spotted Anima*l' can be initiated optionally from '*Check Spotted Animal*'. Next to the main use cases at least one include- or extend use case is used.

b) Give a use case description, including 1 alternative flow, of '*Check Spotted Animal*', following the standard notation (see figure).



Use Case Name
Brief Description
Basic Flow
1. **First step**
2. **Second step**
3. **Third step**

Alternative Flows
1. **Alternative flow 1**
2. **Alternative flow 2**
3. **Alternative flow 3**

P2. Problem Domain Modelling (15)

Build a model of the 'problem' domain presented in the description on pages 2-3 through the use of a UML class diagram. The domain model does **not focus on implementation details**. You should consider using the richness of UML (such as association names, composition- , aggregation-, inheritance relationships and multiplicities) where appropriate.

P3. Design Modelling (20)

A developer of a software development team identified classes with the use of the following CRC cards:

Map	
Responsibilities	*Collaborations*
show location show spotted animals show camp sites ….	GPSController Canvas ....

PanicButton	
Responsibilities	*Collaborations*
Activate alarm ….	AlarmController Canvas ....

Design the initial software structure for the Nature Park Wildlife App, with the use of **1) a UML class diagram and 2) sequence diagram.** The design-level class diagram focuses on implementation aspects. The design should consider the requirements on page 3 and the classes presented by the CRC cards above. For full score all classes, attributes, methods, and associations, association names, cardinalities, and inheritance relationships should be defined, coupling and cohesion is considered. **Procedure**: Create 1 design class model by first constructing 2 sequence diagrams (or the other way around) for 1) checking an animal that was spotted ('Check Spotted Animal') and 2) showing an animals information when clicking it on the map. **The answer should consist of the final sequence- and class diagram.** The diagrams should be consistent.

P4. Design Principles and Patterns (10)

a) Organize your design (class diagram P3) in a layered architecture. Draw classes without the details (attributes/operations). Use proper names for the layers. After introducing layering, it is possible that design principles are violated. Mention which principles are violated and how this should be improved in a next generation of the design. Explain your answer (don't alter the design).

b) One can apply the state design pattern to make further extension of the software easier. Explain how the observer pattern supports the application performance in this particular case.

P5. Behavior Modelling (10)

The panic button can be in different states. When it is pressed too short it stays in 'IDLE' state. In 'IDLE' state, the button is yellow. Pressed for 2 seconds it is in 'LOCKED' state. The color of the button changes into red. Within this state, every 10 seconds the **AlarmController** is used for sending out an alarm message. When the alarm is confirmed by the control centre of the park the panic button turns orange and is in 'WAIT FOR HELP'' state. Pressing 2 seconds again from 'LOCKED' or 'WAIT FOR HELP' sets the panic button back to 'IDLE'.

To model the behavior: 1) consider the panic button as an object. 2) describe the behavior for the panic button by using a UML State Machine Diagram. The state diagram should consist of proper internal (do/) activities, transition triggers and transition effects. It should involve method calls to the relevant objects of the class design of P3. It should be consistent with the answer on P3.