Dipartimento di Elettronica, Informazione e Bioingegneria
Politecnico di Milano

Prof. Elisabetta Di Nitto, Matteo Rossi and Damian Tamburri

20133 Milano (Italia)
Piazza Leonardo da Vinci, 32
Tel. (39) 02-2399.3400
Fax (39) 02-2399.3411

Software Engineering 2 – Written Exam 2 (WE2)

July 1st, 2022

Last Name, First Name
Id number (Matricola)
Number of paper sheets you are submitting as part of the exam

Notes

1. Remember to write your name and Id number (matricola) on each piece of paper that you hand in.
2. You may use a pencil.
3. Incomprehensible handwriting is equivalent to not providing an answer.
4. The use of any electronic apparatus (computer, cell phone, camera, etc.) is strictly forbidden, with the exception of an ebook reader.
5. The exam is composed of 2 parts, one focusing on requirements, and one focusing on design. Read carefully all points in the text!
6. **Total available time for WE2: 1h and 30 mins**

System Description: PoliFlix – a content management platform for next-gen MOOCs@PoliMi

Politecnico di Milano wants to build **PoliFlix**, an online content management platform, connected to an external video streaming system, to support its Massive Open Online Course offer. **PoliFlix** is used by lecturers to provide and classify video contents and by students, who view videos either alone or in a collaborative manner (this is called "stream party" and is explained below).

PoliFlix also gathers user statistics (e.g., most common topics browsed/streamed per user, login time, etc.) and content statistics (e.g., times content was streamed) to enable content management, suggestions and also the proper configuration of the associated streaming system.

PoliFlix should be designed to recommend streaming content to users by topics (e.g., Artificial Intelligence, Public Spaces, Smart Cities). For each user, the topics of his/her interest are *inferred*---namely, they are elicited via an automated *topic modelling* function provided by the platform itself.

PoliFlix lets lecturers upload content on the platform from local storage or link content available from external systems (e.g., Webeep, Webex, YouTube, Vimeo).

Besides the usual individual fruition, **PoliFlix** offers also the possibility to adopt a collaborative video fruition approach (stream party). In a stream party a user acts as the leader and invites other users to view a video together. Each user accepting the invitation views the video from his/her own device and can interact with the other users through a shared blackboard also provided by **PoliFlix**. Video visualization is controlled (start/stop/pause) by the leader for all users. The blackboard allows users to write/delete/modify text.

Part 1 Requirements (6 points)

RASD_Q1 (2 points)
With reference to the Jackson-Zave distinction between the world and the machine, identify the relevant world phenomena for **PoliFlix**, including the ones shared with the machine, providing a short description if necessary. For shared phenomena specify whether they are controlled by the world or the machine.

RASD_Q2 (2 points)
Define the goals for the **PoliFlix** system.

RASD_Q3 (2 points)
Select one of the goals defined in point RASD_Q2 and define in natural language suitable domain assumptions and requirements to guarantee that the **PoliFlix** system fulfills the selected goal.

Part 2 Design (8 points)

DD_Q1+Q2 (2+3 points)
Assuming you need to implement system **PoliFlix** analyzed above:

1. Identify the most relevant components and interfaces and describe them through a UML Component diagram.
2. Provide a brief description of each identified component.
3. For each component, list the operations it provides through its interfaces.

NB: For each operation, you do not need to precisely specify its parameters; however, you should give each operation a meaningful enough name to understand what it does; you can also briefly describe what information operations use/produce.

DD_Q3 (3 points)
Describe through a UML Sequence Diagram the interaction that occurs between the system components when a user sets up a streaming party, invites another user (who accepts the invitation), starts the video, and the other user writes a message on the blackboard.

Solutions

RASD_Q1

World phenomena:
A lecturer prepares some form of content
A user wants to visualize content on a specific topic
A user wants to study together with other friends

Shared phenomena, world-controlled
A lecturer inserts content into system either directly from a local storage or through a link
lecturers classify video contents
A user browses through available contents
A user views a video alone
A user views a video in a stream party
A user acting as leader invites other users to view a video together
A user accepts the invitation
A leader starts, stops and pause a video
Users participating in a stream party read/write/delete modify text in the shared blackboard
A system administrator configures the system by assigning content insertion/classification permissions to lecturers

Shared phenomena machine-controlled
PoliFlix activates video streaming to a user in the normal mode or to all users participating in a stream party
PoliFlix creates a shared blackboard for users in a stream party
PoliFlix gathers information to compute user statistics (e.g., most common topics browsed/streamed per user, login time, etc.) and content statistics (e.g., times content was streamed)
PoliFlix gathers information to infer topics of interest of a certain user
PoliFlix suggests a specific video to a student based on his/her topics of interest

RASD_Q2
G1: Lecturers want to make contents available to students, classified by topics
G2: Users want to browse and visualize contents
G3: PoliFlix owners want users to be associated with their preferred topics based on their viewing habits, and want to use these topics to suggest specific content
G4: PoliFlix users want to collaborate with others while watching contents together

RASD_Q3

(For completeness' sake, the solution lists requirements and assumptions for each goal identified in question RASD_Q2; however, listing only the requirements and domain assumptions for one goal was enough to answer this question.)

G1: Lecturers want to make contents available to other students, classified by topics
R1.0: PoliFlix allows users to login into the system
R1.1: PoliFlix recognizes lecturers as users with special permissions
R1.2: PoliFlix offers to lecturers a way to upload contents
R1.3: PoliFlix offers to lecturers a way to publish contents by specifying the corresponding link
R1.4: PoliFlix offers to lecturers the possibility to classify contents by selecting topics from a list
R1.5: PoliFlix allows lecturers to extend the list of topics by specifying others either as sub-topics of a preexisting one or as brand new topics
R1.6: PoliFlix makes all contents uploaded or made available by link visible to all platform users

A1.1: Lecturers are responsible for ensuring that they introduce appropriate extensions to the list of topics (e.g., they avoid duplicates and define proper topic/sub-topic relationships)
A1.2: Lecturers are responsible for the quality of the content they make available through the platform
A1.3: Lecturers are responsible for the appropriate classification of a content

G2: Users want to browse and visualize contents
R1.0: PoliFlix allows users to login into the system (note that this requirement is common to G1)
R2.1: PoliFlix allows users to browse through the topics list, select a topic, and see all contents associated to a topic
R2.2: PoliFlix allows users to search for a specific content by title
R2.3: PoliFlix allows users to search for a specific content by publisher
A1.3: Lecturers are responsible for the appropriate classification of a content (note that this assumption has been defined for G1 but it is relevant also for goal G2)

G3: PoliFlix owners want users to be associated with their preferred topics based on their viewing habits, and want to use these topics to suggest specific content
R3.1: PoliFlix keeps track of the contents watched by each user and of the corresponding topic
R3.2: PoliFlix infers the topics preferred by users based on the information it has tracked
R3.3: PoliFlix periodically suggests new contents to users based on the inferred preferred topics
A3.1: Users watch contents related to their studies through the PoliFlix platform (they do not use other platforms to watch study-related content, so their preferences are entirely determined by what they watch on PoliFlix)
A1.3: Lecturers are responsible for the appropriate classification of a content (note that this assumption has been defined for G1 but it is relevant also for goal G3)

G4: PoliFlix users want to collaborate with others while watching contents together
R1.0: PoliFlix allows users to login into the system
R4.1: PoliFlix allows users to become leaders for a stream party
R4.2: As stream party leader, PoliFlix allows a user to invite other users
R4.3: PoliFlix allows the invited users to accept/reject the invitation
R4.4: PoliFlix allows a leader to starts, stops and pause a stream party
R4.5: PoliFlix allows users participating in a stream party to read/write/delete modify text in the shared blackboard
A4.1: Users know the Polimi email address of the other users with which they want to have a stream party
A4.2: The external Video Streaming System is able to stream content to different users by guaranteeing a good level of synchronization (e.g., less than 30 seconds).

DD_Q1+Q2

A possible breakdown of the system into components is shows below.

PoliFlix is a traditional three-tier system.

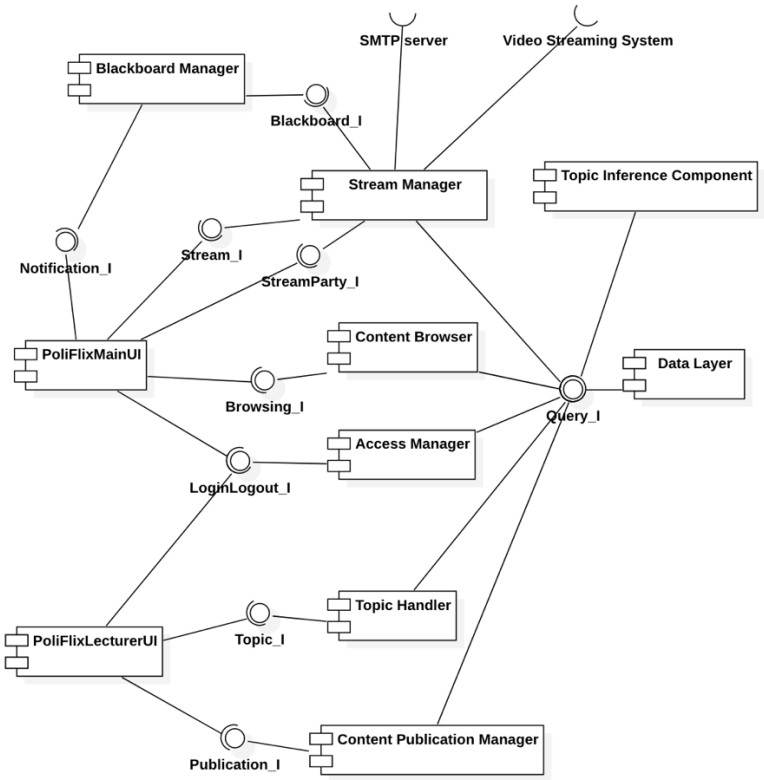
The user interface is decomposed in two parts; one, represented by component *PoliFlixLecturerUI*, is in charge of supporting lecturers in the operations concerning publication and classification of contents; the second one, *PoliFlixMainUI*, supports all users in the operations concerning selection and fruition of contents.

A *Data Layer* component persists data concerning users (lecturers and students, together with their profiles and preferred topics, the topic list, the contents, and data about fruition of contents by users). This could be further decomposed in different parts managing different types of data. The interface offered by the *Data Layer* includes the usual CRUD operations.

The *Access Manager* component is in charge of managing login/logout operations and on associating the right operation types with the two different types of users. *Access Manager* relies on the *Data Layer* for what concerns the information about users and their profiles. *Access Manager* offers the *LoginLogout_I* interface, which includes the usual login and logout operations.

The *Topic Handler* component allows lecturers to update the list of topics. The *Topic_I* interface offers the following operations:
getTopics(), which returns a forest of topics (we could allow also for a non-hierarchical structure like a graph of topics, but its management would require a higher complexity in navigating and updating such a structure);
addNewTopic(), which receives the new topics and its position in the forest;
deleteTopic(), which eliminates a topic and reorganizes the forest.

The *Content Publication Manager* allows lecturers to make contents available and to classify them according to the defined topics. It offers the *Publication_I* interface which includes the following operations:
uploadNewContent(), which receives as parameters the metadata describing a content (publisher, publication data, title, associated topics) and the topic itself, and uploads the new content in the PoliFlix platform;
associateNewContent(), which receives as parameters the metadata describing a content (publisher, publication data, title, associated topics) and the URL of the topic;
publishContent(), which makes a content visible to the students;
addNewTopic(), which receives as parameters a content and a new topic with which the content can be associated.



The *Content Browser* allows users to search for contents. It supports different types of searches: by topic, by publisher, by title. Moreover, it proposes to users the list of potentially interesting contents based on the inferred preferred topics. The corresponding *Browsing_I* interface offers the operations *searchByTopic()*, *searchByTitle()*, *searchByPublisher()*, *getVideosMatchingPreferredTopics ()*, which return the list of contents that fulfill the search query.

The *Topic Inference Component* is activated periodically, typically, when the system is less loaded with requests by the users, and, based on the monitoring information collected in the *Data Layer*, infers new topics of interest for each user. The result of this inference is inserted in the profile of each user.

The *Stream Manager* allows users to visualize contents, by relying on the external *Video Streaming System*, and stores in the *Data Layer* the information about the watched contents by each user. The *Stream Manager* supports both individual content visualization and stream party. For this last process, it relies on an external SMTP server to send invitations to a stream party. Moreover, it activates the *Blackboard Manager* for managing the blackboard in the stream party. This, in turn, relies on the interface offered by *PoliFlixMainUI* to allow the user to be informed of the presence of new messages in the blackboard.
Stream Manager offers two interfaces. *Stream_I* is used for individual content fruition and offering the operations to *start()*/*stop()*/*pauseContent()*. *StreamParty_I* is used for collaborative content fruition. It includes the operations *inviteOthers()*, which receives a list of invited people email addresses, *acceptInvite()*, and *start()*/*stop()*/*pauseContent()*.

The *Blackboard Manager* offers the interface *Blackboard_I*, which provides *setBlackboard()* to set up the blackboard among a stream party participants, *writeMessage()* and *updateMessage()* to write/update messages in the specified stream party board.

DD_Q3