COS301: Mini Project-Phase 1
Functional Requirements

Buzz System

GitHub
https://github.com/RenaldoV/COS301_Group5_a

Khathutshelo Matidza 11072157   Renaldo van Dyk 12204359
Andreas du Preez 12207871   Sean Hill 12221458
Kgomotso Sito 12243273   Hlavutelo Maluleke 12318109
Sboniso Masilela 10416260   Semaka Malapane 13081129

Contents

1 Introduction

The Computer Science Education Didactic and Applications Research (CSEDAR) team of the Computer Science Department of the University of Pretoria have registered a research project called The use of Online Discussions in Teaching (TODT). The aims of this research is to ?nd ways to enhance teaching and improve the learning of students through the use of online discussions. The formation of collaborative communities within student groups has become essential to enhance education. The existence of a forum in which students can express their views and explore what they are learning results in creating a collaborative community in which students can excel. We are faced with the problem of engaging an extremely large number of ?rst year students. Engaging ?rst year students is a di?cult and well-recognised problem, since students tend to feel part of an anonymous mass. As part of the TODT project as a whole, the aim of this COS 301 project is to create an online space where students, teaching assistants and lecturers can engage in activities related to learning the content of our module while applying game concepts to motivate students to increase the quality of their participation and consequently experience deeper learning of the course content.

2 Use Case Prioritization: Reporting

2.1 Critical

6.1 Notification
6.2 View Reports

2.2 Important

6.1 Plagiarism
6.2 Netiquette
6.3 System failure

2.3 Nice-To-Have

6.1 Feedback even when system admin is not online.

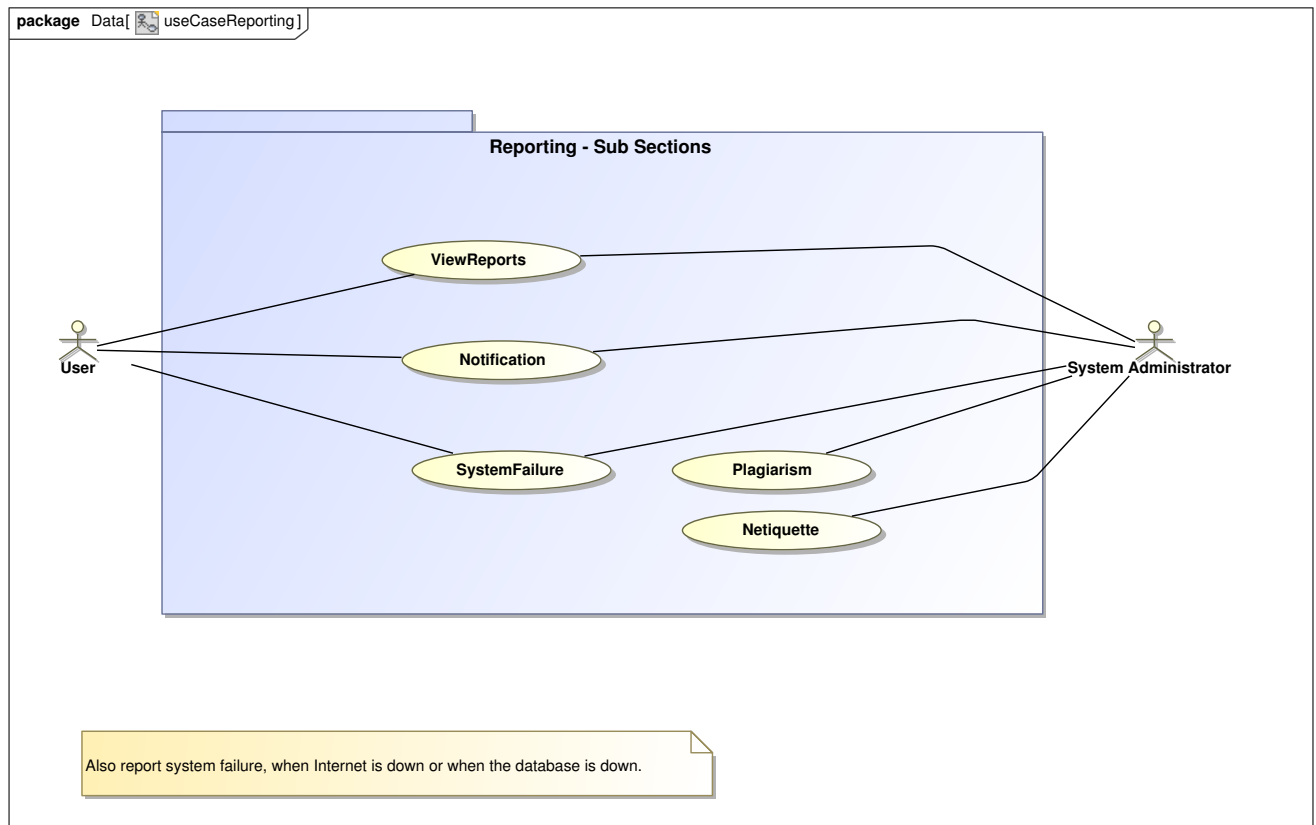# 3 Use Case/Services Contracts: Reporting

3.1 Pre-Conditions

6.1   a) Internet connection and a database required.
6.2   b) Must have logged in to be able to access your notifications.
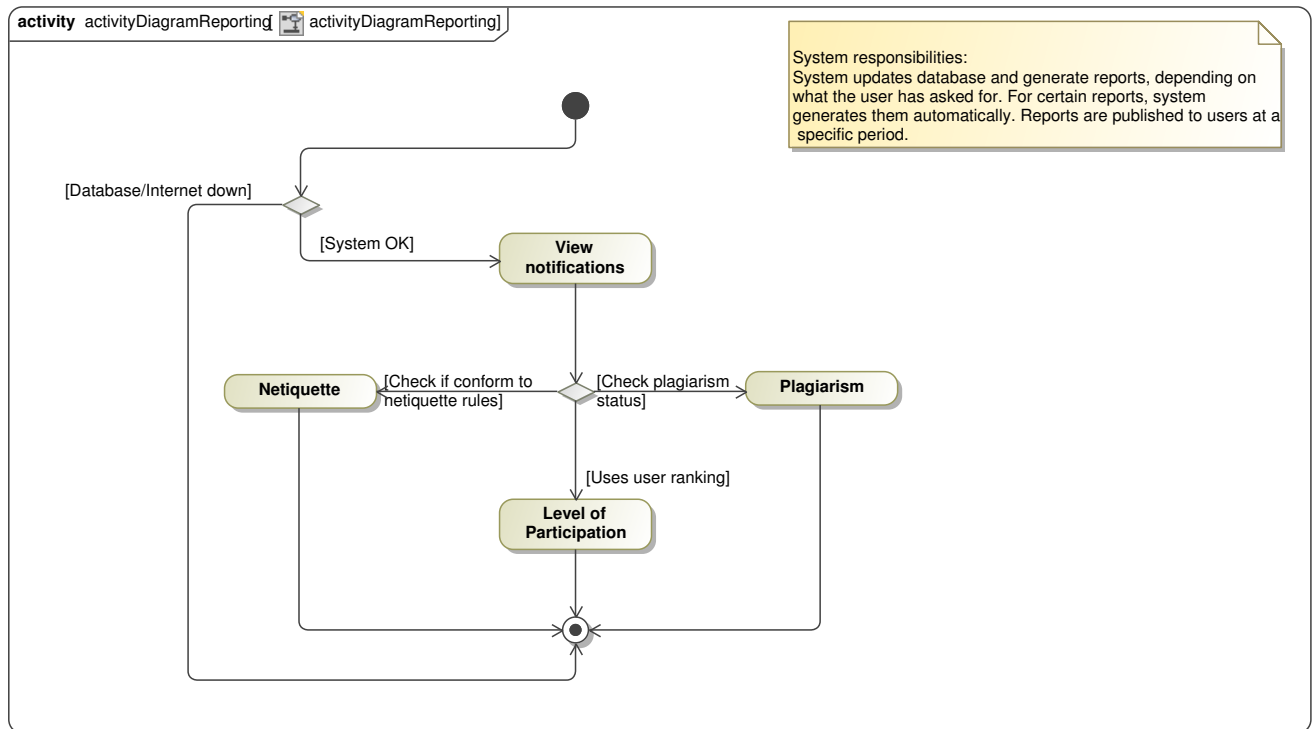6.3   c) System must be running, so to generated updated reports.

3.2 Post-Conditions

6.1   a) System publishes reports to users, on request or when necessary.
6.2   b) Logged in users will be able to see their PLAGIARISM STATUS and if their post comfrom with NETIQUETTE rules.
6.3   a) Activity blog, see what got updated and by who and so forth. (Done by system)

3.3 Request and Results Data Structures

4 Required Functionality: Reporting

package Data[useCaseReporting]

Reporting - Sub Sections

ViewReports

Notification

SystemFailure

Plagiarism

Netiquette

User

System Administrator

Also report system failure, when Internet is down or when the database is down.

5 Process Specifications: Reporting

activity activityDiagramReporting [activityDiagramReporting]

System responsibilities:
System updates database and generate reports, depending on what the user has asked for. For certain reports, system generates them automatically. Reports are published to users at a specific period.

[Database/Internet down]

[System OK]

View notifications

Netiquette   [Check if conform to netiquette rules]   [Check plagiarism status]   Plagiarism

[Uses user ranking]

Level of Participation

6 Domain Model

7 High Level Overview