Bilkent University

Department of Computer Engineering

CS 319 Term Project

ReviewTool

Design Report – Iteration 1

Mustafa Anıl TAŞDAN, Göktuğ ÇAĞIRAN, Cihan Can KILIÇ

Instructor: Eray TÜZÜN
Teaching Assistant(s): Erdem TUNA, Elgun JABRAYILZADE

Contents

1 Introduction

1.1 Purpose of the System

ReviewTool is a desktop application where students can upload the artifacts they produced, post their comments and see reviews posted by their instructors. There is also a peer review system where members can comment on each others' performances.

1.2 Design Goals

1.2.1 Understandability

Since the system will be used by a variety of people, and it's purpose is to reduce the logistics between instructors, TAs and students, it should be easy to use. The system will not achieve it's purpose if users regularly have to contact instructors about questions.

1.2.2 Low Cost

Since we are a very small team consisting of 3 people, it is important that the system is simple and easy to build and maintain.

1.2.3 Reliability

Again, because of limited resources, it is important to build a system that has minimum amount of errors/need of maintenance.

2 Proposed System

2.1 Subsystem Decomposition

The system is constructed in a 3-Layer Architectural Style. Therefore the system is divided into three subsystems. The Database Management subsystem handles data incoming to and outgoing from the client. The Logic Management subsystem acts as an intermediary between the User Interface and Database: holds the current, most recent data received from the database and manages the data going into the User Interface. The User Interface subsystem controls the graphics shown to the user.

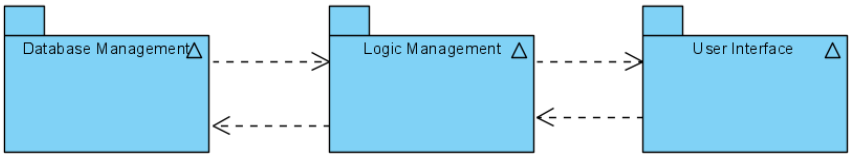

Figure 1: Subsystem Decomposition Diagram

2.2 Hardware/Software Mapping

The application is a Java desktop application, capable of running on any machine that can run the Java Virtual Machine. It connects to a database, the location of which is still undecided, if it is going to be local or remote.

2.3 Persistent Data Management

The system needs to store the following types of data:

- Credentials of users: E-mail, password, user type
- Group Information: Group name, assigned members, assigned assignments, uploaded artifacts, artifact and peer reviews.

Since these are not a big amount of data, they will be downloaded upon login to the system. When a change is made by the user through the UI, the Logic Management subsystem will send the necessary changes to the Database Management system, which will convert the data back to the format used by the database.

2.4 Access Control and Security

There are three types of accounts: Instructor, TA and Student. An Instructor account will have the most authority, will be able to create and manage groups and assignments. The TA account will only keep the ability to view all groups, compared to the Instructor. The Student account will be able to only see his/her own group and will not be able to see peer reviews made by students. All account types will be able to leave artifact reviews, while only a student account can leave peer reviews. Also only a student account can upload artifacts.

2.5 Global Software Controls

The software will utilize event-driven control. When an event is detected through the UI, the event data will be passed from the User Interface subsystem to the Logic Management subsystem. The Logic Management subsystem will execute the necessary operations, like updating the UI or requesting data from the Database Management, based on the event.

The system will follow a centralized design, for two reasons: it is easier to develop, which fits into our "low-cost" design goal, and a possible performance bottleneck is not a concern for an application of this complexity.

2.6 Boundary Conditions

2.6.1 Initialization

Upon startup, there will be no user data stored in the user's computer, there will be only application code. After login, the general user and group data will be downloaded to the system. The whole system starts up at the same time.

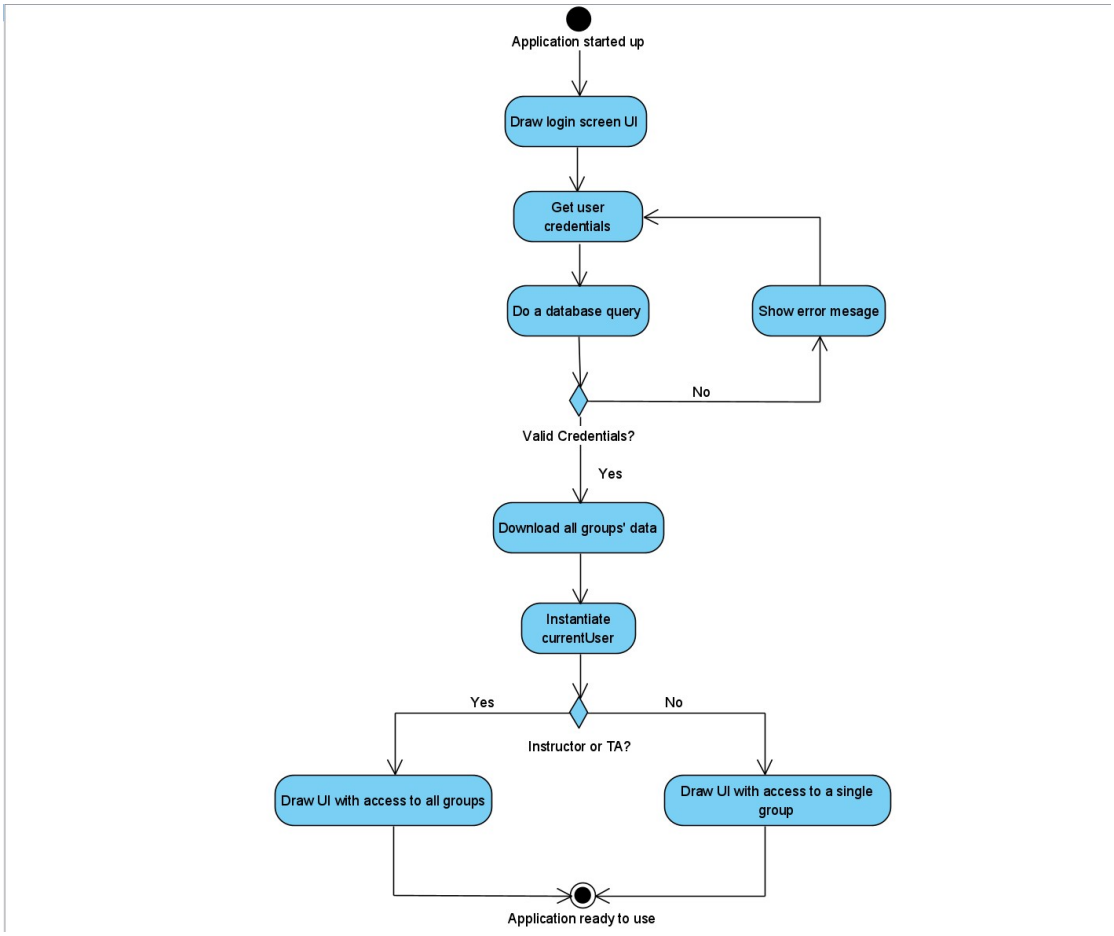

Figure 2: Login System Activity Diagram

2.6.2 Termination

The subsystems' existence are essential for the whole program and they are not individually allowed to terminate. Upon a graceful exit by the user, the downloaded data will be erased from the system. As the updates to the database are communicated at the instant the change is made by the user, no additional transfer is necessary upon termination.

2.6.3 Failure

There are two possible cases for system failure. First is a possible error in the database connection. Here, the application only handles interrupted connections, where it will display an error through the UI. If the data received is corrupted, the application will have no way of knowing/error correcting the said problem. This may lead to incorrect initialization of objects at best, runtime exceptions at worst.

Another possible case is an ungraceful exit from the application, for example a power loss to the computer. The main concern here is that the

application will not have the opportunity to delete the information downloaded from the database, which will be open to access. A possible solution to this is encrypting/decrypting the data.

As these problems conflict with the "reliability" design goal we stated at the start, we do aim to handle these concerns after the initial requirements of the system are met.

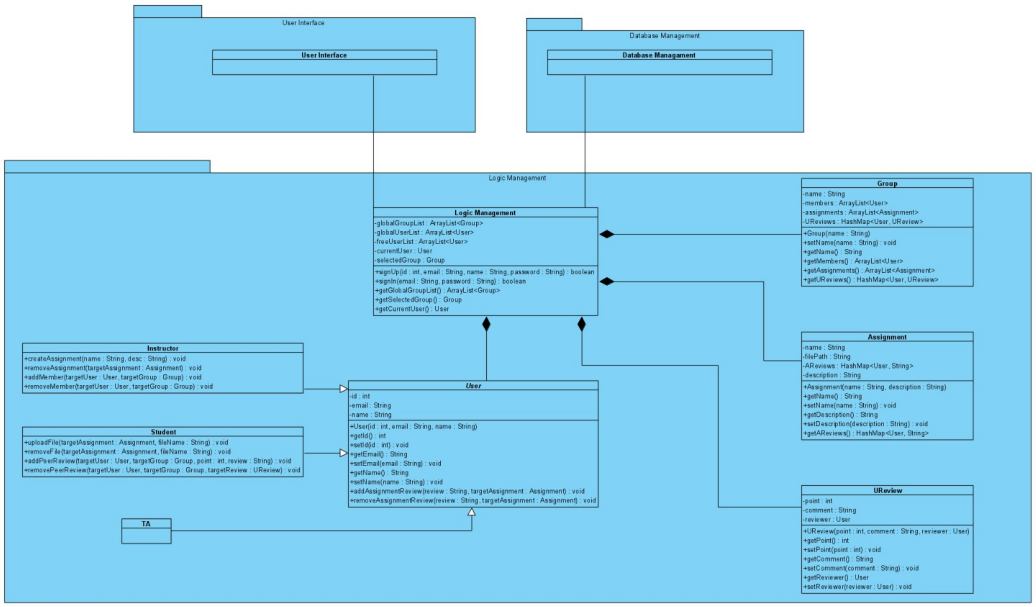

Figure 3: Logic Management Subsystem Class Diagram