Logic In Computer Science

Hamza Najfi - 2023A7PS0673P
Aayush Kumar - 2023A7PS0585P
Shubh Chapra - 2023A7PS0005P
Samarth Kothari - 2023A7PS0646P
Aakarsh Mishra - 2023A7PS0561P

This document provides all the assumptions taken and gives a detailed procedure of how the model works.

Objective : To model a Scholarship Application Portal in UPPAAL which does the following -

- Log in using a student's email ID
- Help a student to apply for a scholarship (once per session)
- Fetch their academic transcript from the server
- View their personal information
- Log out
- Communicate with the Scholarship Approval Committee and the Academic Server

---

Components : The model contains 4 templates as listed below -

1. Student : This template helps a student in applying for a scholarship, asking for their personal information and/or their academic transcript.
2. Portal : This template does all the information retrieval and acts as a communication channel between ‘student’ and all the other templates.
3. Academic_Server : Fetches and relays the transcript back to the Portal template
4. Scholarship_Approval_Committee : Processes a scholarship application and decides whether to grant a scholarship (either MCN or Merit) or not based on a “CGPA” criteria.

Working :

- A student attempts to login to the scholarship portal using their **email** and **password**, which is **verified** by the portal. If incorrect, the student is **returned back** to the logged_out state and would be able to sign_in again.

- After a **successful** sign_in, the student is taken to the **homepage** of the portal, where he has these options - **fetch_transcript, fetch_personal_info and apply_scholarship.**

- Working of each option is described below:
    - Fetch_transcript - The student request is processed by the portal and relayed to the **academic_server**. The server **processes** the request, **fetches** the transcript

and **sends** it to the portal where it can be viewed by the student.

- Fetch personal info - The student requests to **view** his **personal info** to the portal and the portal **displays** the required info to the student.

- Apply scholarship - The student **applies for a scholarship.** A boolean variable acts as a guard and is updated as soon as a successful application is received by the portal. The variable ‘appld’ changes to 1 which prevents further applications by that student which has already applied. The student can **choose to apply** for one of two scholarships - **Merit and Merit-cum-Need (MCN)**. The request is received by the portal which relays it to the **scholarship approval committee**. The committee **receives** the request and **verifies** the student’s CGPA. The merit scholarship has a **cutoff** of **9**, while the mcn scholarship has a **cutoff** of **7** CGPA. The student’s CGPA should be greater than or equal to the cutoff of the scholarship that he has applied to. The committee verifies this and if the student’s CGPA is adequate, **approves** the scholarship request, otherwise it **rejects** it. This decision is then sent back to the portal which then goes to either the approved state or the rejected state.

- After going through any of the options, the student can **return** to the homepage and either access the options again (with the exception of applying for scholarship) or they can

log out of the portal. A student can log in once again once they have logged out.

Assumptions : We have taken the following assumptions and listed the basis for them -

- A student has the credentials - their email ("f20xxXXXX@pilani.bits-pilani.ac.in") and password which is a four-digit number. The correct password is arbitrarily defined to also be a four-digit number (1234).

- The MCN mode of scholarship does not in fact take into account the family income of the student, unlike the real SWD scholarship portal, and is purely dependent on CGPA (however like the real portal the CGPA cutoff is lower than the merit scholarship). However this can easily be remedied by a simple guard check and an extra income variable in the global declarations.
- As per the question, every successful login is made by a unique student.

Safety Checks : Following checks have been done on the model

➔ A[] (Student.Scholarship imply !appId)
  ◆ Whenever a student has submitted scholarship application implies that the student had not applied yet.
➔ A[] !deadlock
  ◆ No Deadlock ever occurs.

- ➔ A[] !(Student.Merit_cum_Need_Scholarship && Student.Merit_Scholarship)
  - ◆ A student can never possess MCN and Merit at the same time.
- ➔ A[] (Student.Logged_out imply not portal.HomePage)
  - ◆ Globally, if a student has logged out then he is not on homepage.

Liveness Check :

- ➔ A<> (Student.Home_Page imply Student.Home_Page)
  - ◆ When a student is at the home page, he can eventually return to the home page.
- ➔ E<> (cg>9 imply Portal.Approved)
  - ◆ Eventually in some path, A student with CGPA > 9 gets an approved scholarship.
- ➔ A<> (Student.Fetch_Transcription imply Academic_Server.Fetched_Transcription)
  - ◆ Whenever a student requests to fetch transcription, the Academic server eventually fetches the transcription.
- ➔ A<> (Scholarship_approving_Committee.Approved imply portal.HomePage)
  - ◆ For all paths, if Committee approves Scholarship request then student eventually reaches portal homepage.
