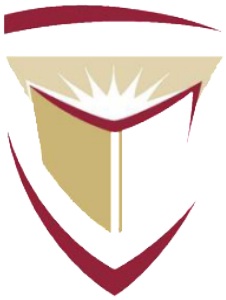


UNIVERSITÉ Concordia UNIVERSITY

Concordia Institute for Information Systems Engineering

INSE 6250 Quality Methodologies for Software

Summer 2021

COVID-19 Fact Checker Website Model Checking Using UPPAAL

Dr. Jamal Bentahar

Shivam Pandya 40156978 Quality System Engineering, Gina Cody School of Engineering and Computer Science, Canada.

Shamiran Asslan 40108263 Quality System Engineering, Gina Cody School of Engineering and Computer Science, Canada.

Date: June 28, 2021

Abstract

Model Checking is a method of checking the correctness of the finite-state model. In other words, it helps to check whether the proposed system model will satisfy the given requirements or not. For this project, we are going to develop a model for Covid-19 Fact checking website and perform model checking for it. The system will be modeled by using update, guard, and sync conditions. The system model will be structured by using the CTL formal language. Moreover, the formal specifications will be checked in the UPPAAL software.

1- Introduction

The Internet is considered as the ocean of Information; however, the authenticity of the information is still questionable as everyone can post it and edit it. Social media exacerbates this situation. In certain situations, like Covid 19 Pandemic, where everyone is frightened and feels unsafe, rumors and misinformation can lead to serious consequences. Thus, it is evident that for the current scenario, the fact checking website for Covid-19 information should exist [1],[2]. This fact checker must rely on the authentic source of information like government websites and research papers. We Have formulated a model for this system and check the system using UPPAAL to know whether it satisfies the given requirements or not.

2- Scope

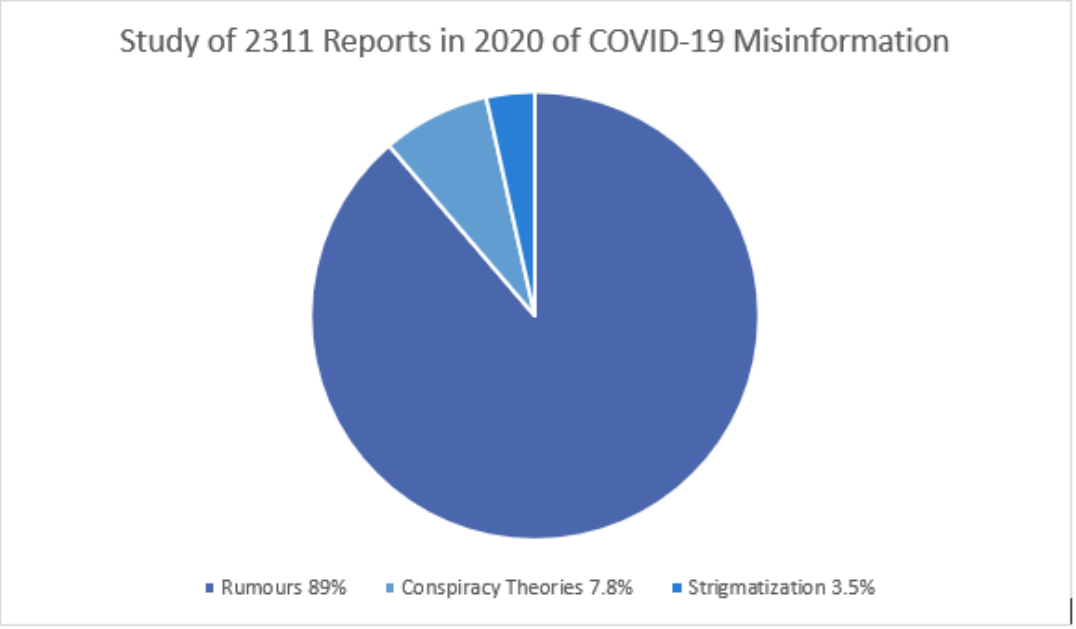
The scope of the project is to apply verification and validation of web applications and check the system robustness and system working properly according to the developed functions and properties. The idea of doing this project is to find how the system is behaving in the different states and what will be the current scenario of the system including further improvement of the whole system.

3- Problem Description

It is evident that misinformation and rumours have increased since the spread of social media. During the COVID-19 pandemic, we face the challenge of an overabundance of information related to the virus. Some of this information may be false and harmful. Incorrect information spreads widely and at a speed [3] that makes it more difficult for the public to identify verified facts and advice from trusted sources, such as their local health clinics.

For instance, in Nigeria, they found several cases of chloroquine overdose after news reported the effectiveness of the drug. In the USA, they panic buying groceries, fuel, paper products after rumours about a lockdown. In Greece, incorrect information caused a drug shortage. A study [4] showed that a total of 2049(89%) out of 2311 reports were classified as rumours,182 (7.8%) were conspiracy theories, and 82 (3.5%) were classified as stigmatization of an affected group.

However, everyone can use the COVID-19 Fact check website. This website is an effort to collect and communicate the most relevant and useful science for information regarding coronavirus disease to obtain valid and accurate information.



4- UPPAAL

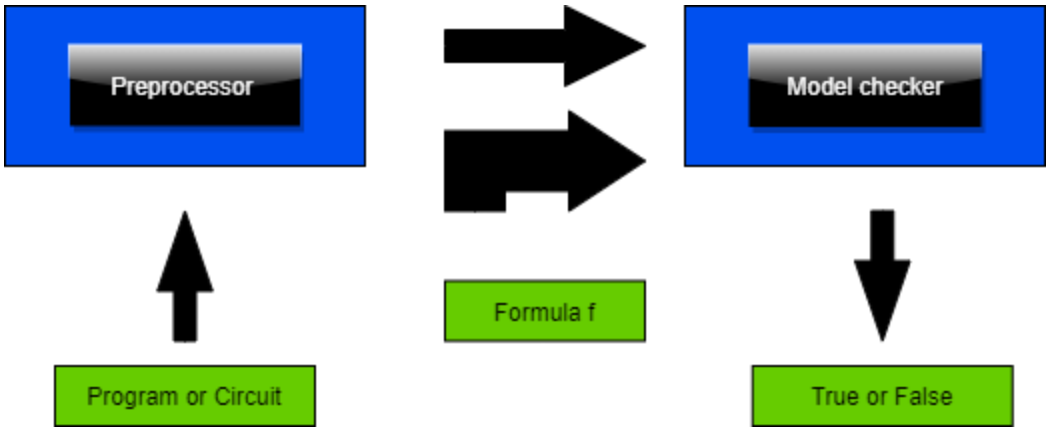
Uppaal is a free toolbox for non-commercial applications in academia only for modeling, validating and verifying real-time systems designed as networks of timed automata, extended with data types of bounded integers, arrays, etc [5]. The current official release is Uppaal 4.0.15 (November 18, 2019) which will be used in this project. The software runs under Windows, Mac OS X, AND Linux. Advantages of Uppaal are the graphical user interface and the short learning curve. The Uppaal graphical user interface involves three main parts, reachable via three tabs in the main window: the system editor, the simulator, and the verifier. Each tab has a different attribute as an editor can be used to construct models, the simulator where system behavior can be simulated, and the verifier tab which system behavior can be analyzed [6]. Moreover, different languages like CTL (Computation Tree Logic) and LTL (Linear Temporal Logic) are used.

5- Model checking

Model checking is a powerful technique for verifying temporal properties of finite state concurrent systems. In model checking, specifications about a system are expressed as (temporal) logic formulas, and efficient algorithms are used to traverse the model defined by the system and check if the specification holds or not [7]. The processes of model checking are Modeling, specification, verification.

Modeling is the first step that can be used to convert a design into a formalism accepted by a model checking tool. In other words, it is simply a compilation task. Furthermore, a design

modeling might require eliminating unnecessary details due to limitations on time and memory. Specification is given as a logical formalism for hardware and software systems that use temporal logic which can help how the behavior of the system develops over time. Verification is completely automatic. However, in practice, it involves human assistance to analyze the verification results and in negative results, the user may analyze the error trace to modify the system and reapply the model checking algorithm. The figure shows the structure of a typical Model checking system. A preprocessor extracts a state transition graph from a program or circuit. The model checking engine takes the state transition graph and a temporal formula and determines whether the formula is true or not [8].



6- COVID-19 Fact Check Website Model

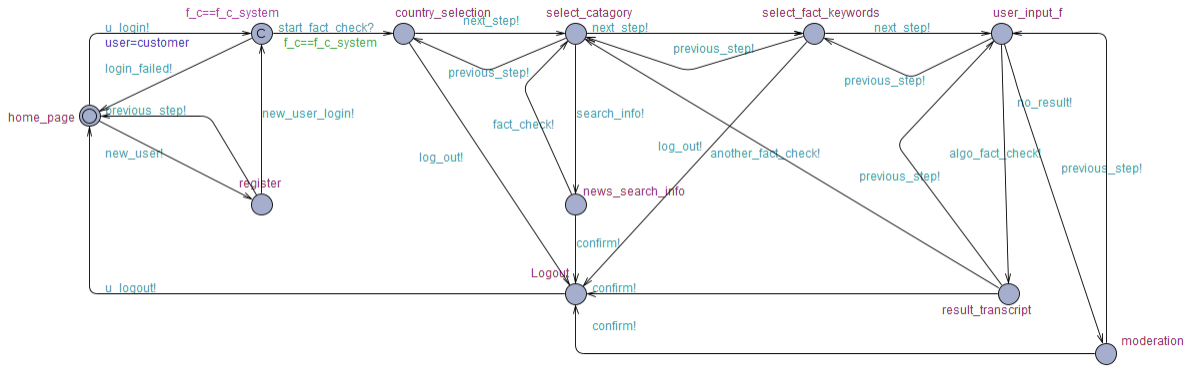
The kripke model of a Covid 10 fact check website will be checked by the various properties that are satisfied by that system. This system model consists of several states and transitions from one state to another or many. Each state represents the webpage and transition represents the movement/activities that the user can do on the particular page.

Users can access this Covid 19 fact check website from any corner of the world. They can access any fact related to Covid 19.

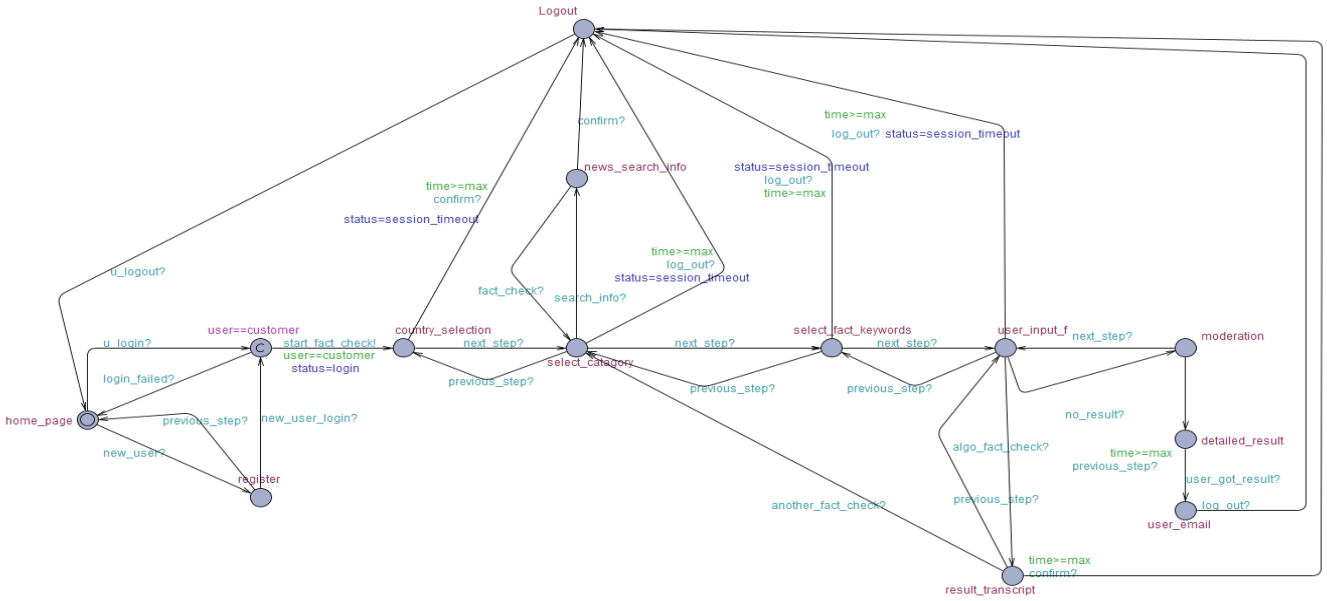
There are main tabs in UPPAAL:

1. Editor:

   Editor is the place where we have to draw a system automaton. It has options for system and variable declaration. In system editor, we structured the system and defined the states and transitions between various states. For our system, we need to have two models; the first is User model and second is Server Model.



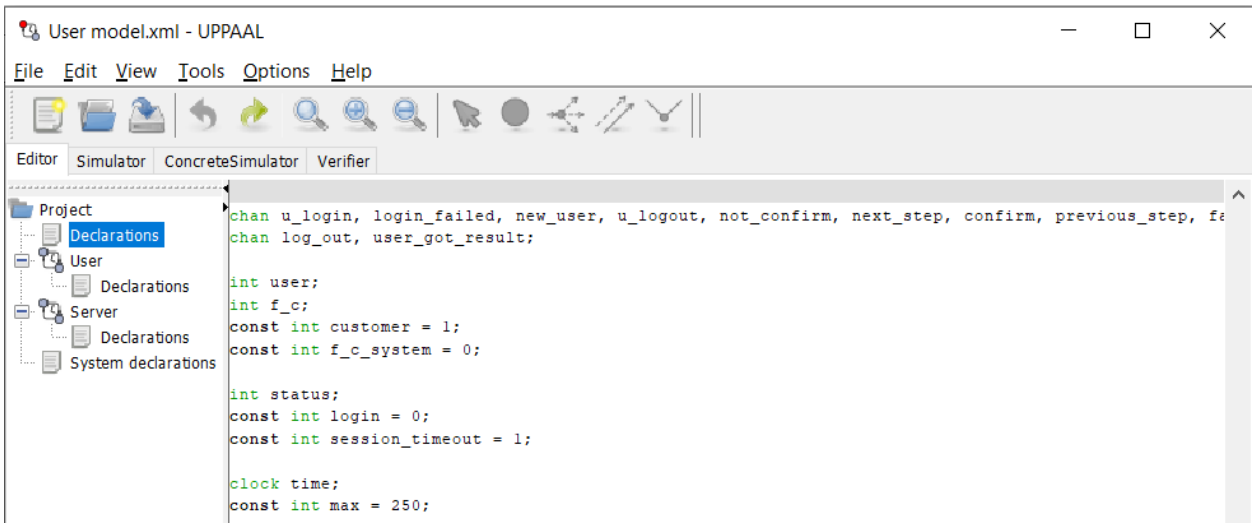
The homepage is the initial and final state of the whole system. Users have to login to the system for any fact check or to see the authentic news. New user has to first register and then login to the system. If any user logs out between any state of the website, the user will navigate to the home page.



In the next state, User has to select the country and then select the fact category. These options help the algorithm to narrow down the search option. Moreover, there are two kinds of fact checking processes that will happen in this model. The first one is an algorithmic process and the other is a manual fact checking process(moderation). The system flow is designed such that fact will be checked by the algorithm first. If the algorithm couldn't find a proper result, thereafter it will go under moderation, where a technical team will search the authenticity of the fact.

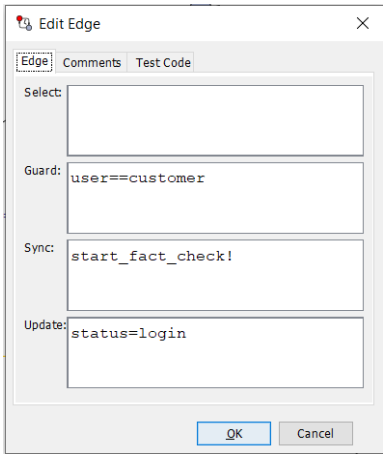
Majority of the channels and states of User and Server models are the same, because they have to be in line to make a functioning website. However, there are some additional channels and states in the Server model. Under the Project Declaration, we declare all channels and variables of the system.

In declaration, we have also listed out the variables and clock variables for the log out. The user will be automatically session timeout if it stays inactive on specific pages more than 250 sec.



In the system model, there are some options that are essential to know about. They are;

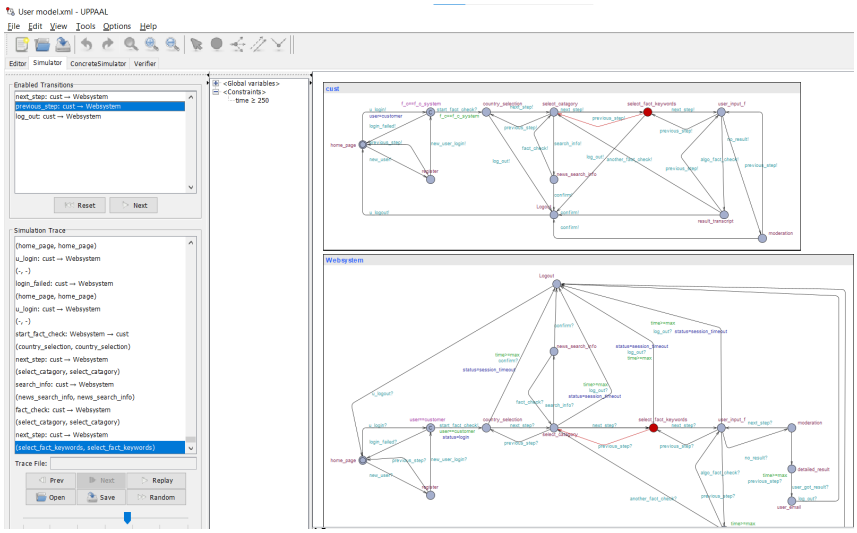
- Location: It represents the webpages of the websystem.
- Channel: It helps to depict the transition from one state to another. It is a medium through which one state communicates with another.
- Sync (Synchronization): This function shows the sync action between two systems, here in this case, the two systems are Websystem(Server) and User.
- Guard: It checks special conditions that decide whether to update the clock or integer variables.
- Update: It helps to show change in the state of the system.



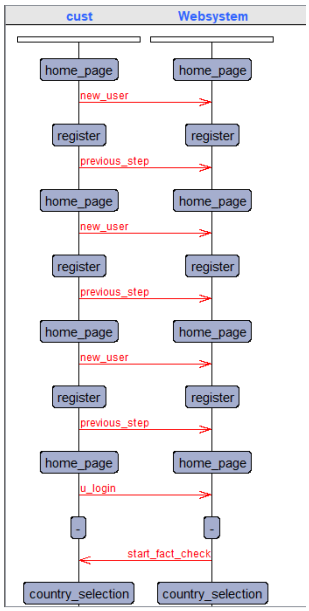
2. Simulator:

Simulator checks the model behaviour. Once the error free model is structured, the simulator checks for the deadlock by simulating messages between different states. The simulator checks the system transitions of the User and Server model. It also provides a visual representation of the transmission. Moreover, we can check the transition of a particular one state and also simulate it randomly. For this, the Random option is useful. In the simulation, the red dot on the model illustrates the transmission of messages from one state to another. So, if there is any deadend in the model, it will stop transmission and you can see which state has the issue.

Simulator also shows the trances of the system and the updation of variables with respect to transitions. We can save or review the trace files. The speed of simulation can be managed by the speed option. This figure is a demo of the simulator tab;



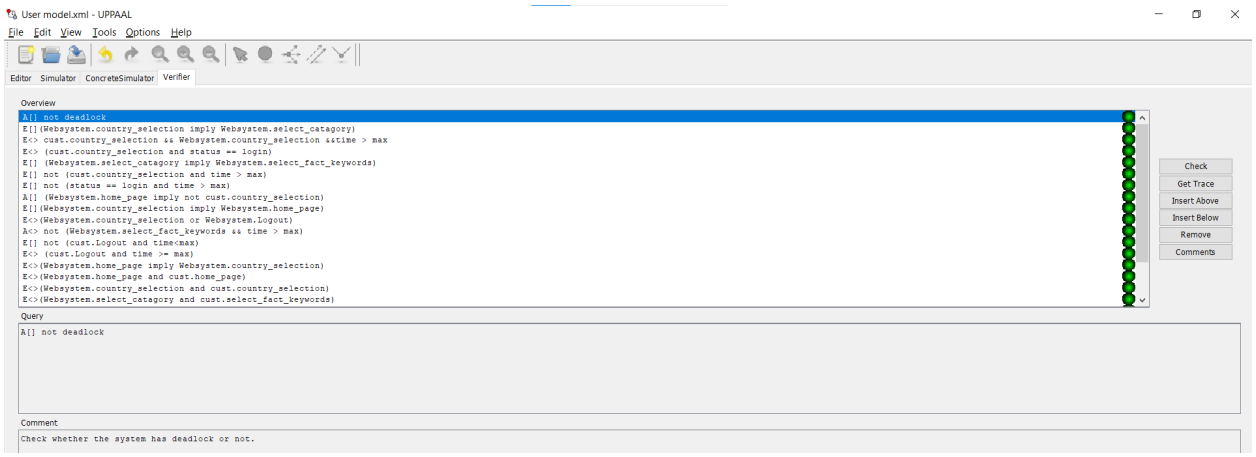
The following figure shows the transmission of messages between the customer and Websystem.

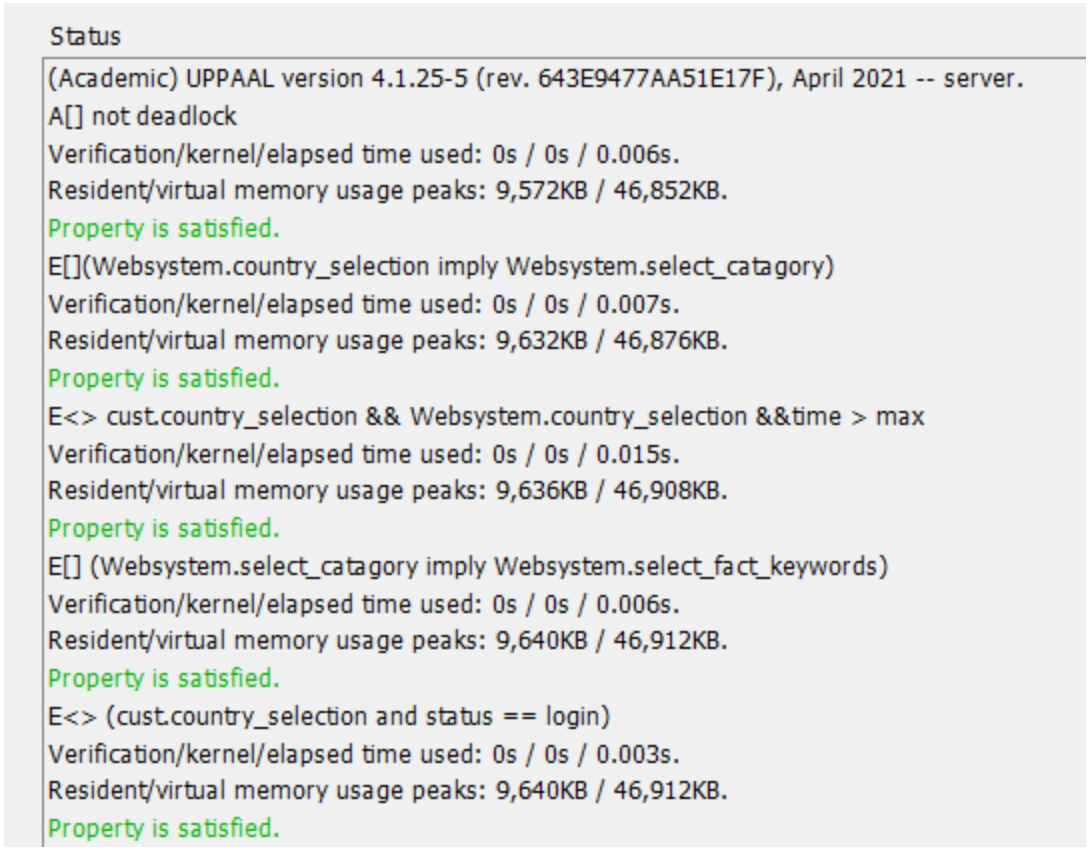


3. Verifier:

In the Verifier, first we have to write the queries. Verifier checks whether the queries are satisfied in the model or not. Queries are properties or specifications that have to be satisfied

by the model. The most important property is checking dead ends in the system. For this we have used the “A<> not deadlock” equation and it is satisfied in both of the models.





The green button illustrates that the property is satisfied, the grey shows that the property is not checked and the red colored button depicts that the property is not satisfied in the model.

Property Specification:

No.	Properties (Queries)	Comments
1.	A[] not deadlock	It checks whether the system has deadlock in all paths or not.
2.	E[](Websystem.country_selection imply Websystem.select_catagory)	This checks that there exists a path in which all states are reachable from country_selection.
3.	E<> cust.country_selection && Websystem.country_selection && time > max	It checks that there exists a path in which the clock will be triggered (session timeout)when time is greater than max value.
4.	E<> (cust.country_selection and status == login)	This property checks status is logged in and after checking login credentials.
5.	E[] (Websystem.select_catagory imply Websystem.select_fact_keywords)	This checks that there exists a path in which all states are reachable from select_catagory to select_fact_keywords.
6.	E<>(Websystem.Logout and cust.Logout)	It covers Logout must at least once in future in both Websystem and Cust.
7.	E[] not (cust.country_selection and time > max)	It checks that session timeout will not occur if User is active.
8.	E[] not (status == login and time > max)	It checks that status will not be on the path if the user is not active.
9.	A[] (Websystem.home_page imply not cust.country_selection)	This property checks that the user can't bypass the home_page without the login credentials checking process to start the fact check directly.
10.	E[](Websystem.country_selection imply Websystem.home_page)	It checks if there exists a path which is reachable to home_page from Country_selection.
11.	E<>(Websystem.country_selection or Websystem.Logout)	It checks there exists a path where users can log out in future from the Country_selection Page.
12.	A<> not (Websystem.select_fact_keywords && time > max)	It checks in all paths that if the user is active session time out will not occur.
13.	E[] not (cust.Logout and time<max)	It checks that session time out will occur when user is not active.
14.	E<>(Websystem.home_page imply Websystem.country_selection)	It checks there is a atleast a path where user can go to home_page from Country_selection.

15.	E<>(Websystem.home_page and cust.home_page)	It checks that user will reach to home_page at least once in the future.
16.	E<>(Websystem.country_selection and cust.country_selection)	It checks that user will reach to country_selection page at least once in the future.
17.	E<>(Websystem.select_catagory and cust.select_fact_keywords)	It checks that user can go to select_catagory from select_fact_keywords at least once in future.
18.	E<>(Websystem.user_input_f and cust.user_input_f)	It checks that user will go to user_input_f at least once in future.
19.	E<>(Websystem.result_transcript and cust.result_transcript)	It checks that user will go to result_transcript at least once in future.
20.	E<>(Websystem.moderation and cust.moderation)	It checks that user will go to moderation at least once in future.

7- Conclusion:

The findings of this project were how to model a real-time finite system in UPPAAL and check whether the model satisfies certain properties that are given by the customers. UPPAAL has a great user-interface and allows users to check whether the model satisfies the given requirements or not.

Covid 19fact checking website model doesn't have any deadlock in the model. And it satisfies the 20 properties. This system can handle the user inputs and also provide accurate results. The algorithm fact checking and manual fact checking are the unique features of this website. This website allows users to check covid 19 related facts and they will be aware about the authentic information.

8- References

[1] E. Lurie, “The Challenges of Algorithmically Assigning Fact-checks: A Sociotechnical Examination of Google’s Reviewed Claims,” p. 98.

[2] “Fact Check Tools.” https://toolbox.google.com/factcheck/explorer (accessed May 23, 2021).

[3] A. Tentolouris, I. Ntanasis-Stathopoulos, P. K. Vlachakis, D. I. Tsilimigras, M. Gavriatopoulou, and M. A. Dimopoulos, “COVID-19: time to flatten the infodemic curve,” *Clin. Exp. Med.*, vol. 21, no. 2, pp. 161–165, May 2021, doi: 10.1007/s10238-020-00680-x.

[4] M. S. Islam *et al.*, “COVID-19–Related Infodemic and Its Impact on Public Health: A Global Social Media Analysis,” *Am. J. Trop. Med. Hyg.*, vol. 103, no. 4, pp. 1621–1629, Oct. 2020, doi: 10.4269/ajtmh.20-0812.

[5] “About | UPPAAL.” https://uppaal.org/ (accessed May 23, 2021).

[6] F. Vaandrager, “A First Introduction to Uppaal,” p. 33.

[7] O. Grumberg and H. Veith, *25 Years of Model Checking: History, Achievements, Perspectives*. Springer, 2008.

[8] E. M. C. Jr, O. Grumberg, D. Kroening, D. Peled, and H. Veith, *Model Checking, second edition*. MIT Press, 2018.

[9] INSE 6250 Lecture notes, by Dr. Jamal Bentahar.