Deliverable 2-3

Cpt_S 583 Software Quality

Electronic voting system

Lieh-Hsuan Chen
Hung-Wei Lee
Lei Shang

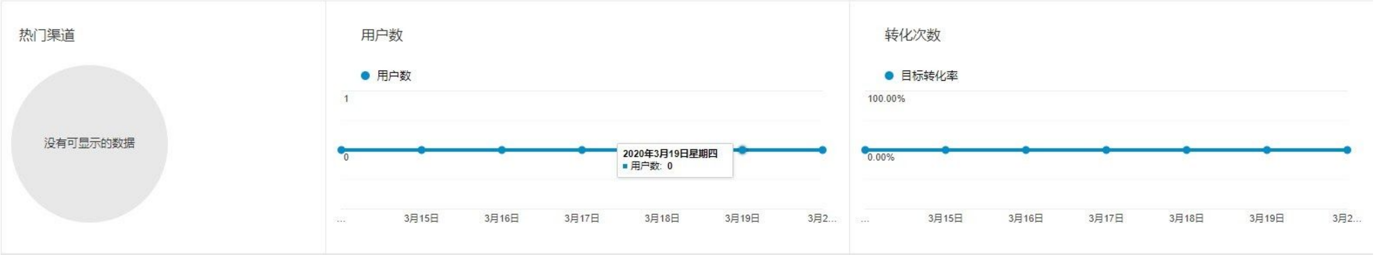
March 3, 2020

Product Quality Metrics:

Quality: Availability
Goal: The system shall work continuously in 3 days
Metric:

item	Availability
name	The work hours of the system.
Costs	working hours can be observed in Google Analytics
Target Value	3 days
Tools	Google Analytics
Application	Metric is used for measure the availability of the system
Data Items	detecting time
Computation	period of time

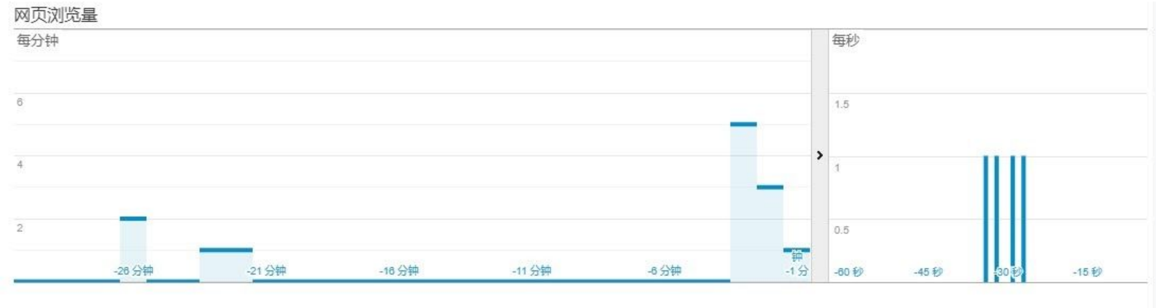
Validation Results: In order to test availability of the website, we monitored our project with Google Analytics and catched the report of performance. Consequently, the website keep working and provide normal functions to the users. There is no memory leakage problem which will cause website crash.



Quality: Reliability
Goal: The error rate for operations should be below 1%.
Metric:

item	Reliability
name	The errors detected in the system
Costs	error can be continously detected by circle CI
Target Value	<1%
Tools	circle CI
Application	Metric is used for measure the error rate of the system
Data Items	bugs
Computation	number of bugs and error rate

Validation Results: According to the Google Analytics, the website provide user registration, and login functions to users, which exchange data with backend of database. In addition, the website also provide users to vote their candidate, the records also stored in the database. Within the interaction between frontend and backend, the website works normally in every minute.



Quality: Robustness
Goal: There should be zero errors that are a function of input data.
Metric:

item	Robustness
name	The wrong input when doing CRUD with database
Costs	wrong input of format data in the system
Target Value	0
Tools	Fronted detection & backend detection

Application	Metric is used for measure the wrong input in the system
Data Items	wrong input data or wrong format data
Computation	number of wrong input data

Validation results: All input fields will be through a double validation system. Firstly, the frontend will validate user input before sending to the backend. Then, the backend will validate data before sending it to the database. Thus, the robustness will be ensured in this project.

Quality: Learnability
Goal: A new user should be able to use the system effectively in an hour.
Metric:

item	Learnability
name	The time that user familiar with this system
Costs	time cost can be obtained by survey or poll
Target Value	<= 1 hrs
Tools	Frontend UI, KML computation
Application	Metric is used for measure the time that user faimilar with this system
Data Items	time
Computation	counting

Validation results: There are obvious navigation bars besides the page, users will clearly know how to operate the system and direct to other pages. The design of this website is also very simple, there are no confusing components and icons.

Quality: Usability
Goal: There should be no more than three complaints from users when trying to learn the system.
Metric:

item	Usability
name	The satisfaction of the users.
Costs	time costs can be obtained from users
Target Value	3

Tools	test by tester
Application	Metric is used to measure how satisfaction is the system
Data Items	detecting time
Computation	counting

Validation results: The users feel good at operating the website, the UI is clear and simple.

Quality: Efficiency
Goal: The system should return search results in less than one second at the 95th percentile.
Metric:

item	Efficiency
name	The time cost that system response
Costs	time costs can be obtained from testing tools
Target Value	3 sec
Tools	google analytics
Application	Metric is used to measure how efficiency is the system
Data Items	detecting time
Computation	counting

Validation results: By Google Analytics report, the time cost of system response is in one second and accurate for a day. The system can exchange the data with the backend correctly in every second.

Quality: Security
Goal: The token stored in the website should be cleared in a day.
Metric:

item	Security
name	The token that is stored on the website
Costs	costs can be obtained from testing tools
Target Value	3
Tools	Kali Linux
Application	Metric is used to measure how security is the system

Data Items	security problems
Computation	average security problems

Validation results: In this project, when the user login the system, the backend will send an encrypted token to the user, and store it in the frontend. There is an expired time in the token, if user login and stay exceed the expired time, the system will automatically logout. In this case, we can ensure security of login at different locations.

Quality: Portability
Goal: The system should work with a variety of environments with less than an hour of configuration needed.
Metric:

item	Portability
name	The hours of system configuration
Costs	time costs can be obtained from testing tools
Target Value	3 hours
Tools	Docker & kubernetes
Application	Metric is used to measure how convenience can the system portable
Data Items	detecting time
Computation	average system portable time

Validation results: The system will be containerized with docker, such that the frontend, backend, database will be three microservices. In that case, all services can be configured in different environments quickly.

Process Quality Metrics:

Quality: Maintainability
Goal: The new features enhanced in the system required by the customers should be done in less than three weeks.
Metric:

item	Maintainability
name	The satisfaction of the customers.
Costs	time costs can be obtained from testing tools
Target Value	3 hours

Tools	circle CI
Application	Metric is used for measure the time cost of add a new features
Data Items	detecting time
Computation	average new features time

Validation results: The new features like the business logic will be developed by the Visual Studio Code and also keep it compatible with the changing software environments such as hardware and operating systems. In this case, all new features can be done quickly to reduce the costs and meet the customers' requirements.

Quality: Testability
Goal: The test should detect the bugs in the system in less than half days.
Metric:

item	Testability
name	The time cost of test detecting
Costs	data can be obtained from testing tools
Target Value	30 mins
Tools	REST assured
Application	Metric is used for measure the time cost of test detecting
Data Items	detecting time at code inspections
Computation	counting

Validation results: Exhaustive testing is easily achievable and practically better if applied isolatedly for every component on all possible measures that add to its quality instead of trying to test the finished product with use-cases that tries to address all components. In this case, we should build the components highly testable as much as possible to meet the goal of testability.

Quality Goals:

Product Quality	**Quality Goals**	**Quality Metrics**	**Verification and Validation Results**
Availability	The system can work continuously in	The system should have less than 1%	The details have been posted above.

	3 days	error rate on the days.	
Reliability	The system needs to provide correct operations for the user.	The operation of error rate should be less than 1%.	See details above.
Robustness	the input data should have no error for the system.	The data entered is correct and sent to the backend and database to storage. Also, the system should control there is no error of input.	See the detailed description above
Learnability	Users can be familiar with the time cost of website operation	A new user can be able to use the system efficiently in less than 24 hours.	See the detailed description above
Usability	The system should be highly user-friendly.	There are less than 3 complaints for the user who tries to learn this system.	See the detailed description above
Efficientcy	The system should return the results of searching.	After searching, the system should be able to return the results in less than 1 second (95%)	See the detailed description above
Security	The system should clear the token for security.	The system should guarantee that cleaning the token per day.	See the detailed description above
Portability	The system should be able to work in various of environments.	The system should be configured in less than 1 hour.	See the detailed description above
Process Quality	**Quality Goals**	**Quality Metrics**	**Verification and Validation Results**
Maintainability	The new features enhanced in the system required by the customers should be done in	The satisfaction of the customers.	See the detailed description above

	less than three weeks.		
Testability	The test should detect the bugs in the system in less than half days.	The time cost of test detecting.	See the detailed description above.

Reference:

[1] Matt Crane, Andrew Trotman, and Richard O'Keefe. 2013. Malformed UTF-8 and spam. In Proceedings of the 18th Australasian Document Computing Symposium (ADCS '13). Association for Computing Machinery, New York, NY, USA, 101–104. DOI:https://doi.org/10.1145/2537734.2537746

[2] Christopher Henard, Mike Papadakis, Mark Harman, Yue Jia, and Yves Le Traon. 2016. Comparing white-box and black-box test prioritization. In Proceedings of the 38th International Conference on Software Engineering (ICSE '16). Association for Computing Machinery, New York, NY, USA, 523–534. DOI:https://doi.org/10.1145/2884781.2884791