UTM UNIVERSITI TEKNOLOGI MALAYSIA
INTERNATIONAL JOURNAL OF INNOVATIVE COMPUTING
ISSN 2180-4370
Journal Homepage : https://ijic.utm.my/

Interactive Course Registration System with Visualized Timetable

Amirul Faiz Ahmad Puad[1], Mohd Razak Samingan[2] & Muhammad Khatibsyarbini[3]
Faculty of Computing, Universiti Teknologi Malaysia
81310 UTM Johor Bahru, Johor, Malaysia
Email: amirulfaiz98@gmail.com[1]; mrazak@utm.my[2]; khatibsyarbini@utm.my[3]

Submitted: 17/1/2024. Revised edition: 28/3/2024. Accepted: 12/5/2024. Published online: 25/11/2024
DOI: https://doi.org/10.11113/ijic.v14n2.504

***Abstract*—Course Registration System (CRS) is a web-based application that can be accessed by any platform and device as long as there is the internet that can be connected to the web service. The target user of the system is the student and staff of the school of computing Universiti Teknologi Malaysia (UTM), Johor Bahru. The CRS is a system that can provide a feature that helps the student to perform the course registration process sufficiently. The CRS system is developed through a hybrid agile waterfall methodology. The hybrid methodology provide systematic documentation for the project while offer flexibility during its development phase. A visual timetable generated and able to view the list of course mate in the registration process are major features that CRS provide. These features are the basic things that should be able to be implemented in course registration for it easier for the student to identify the clash course taken and make their study plan. At the end the CRS system able to provide a complete feature that make students register for a course with less burden and less complexity through the visualized table feature.**

***Keywords*—Web-Based Application, Online Enrollment System, Hybrid Agile Waterfall Methodology**

I. INTRODUCTION

Every student especially at the university level needs to register for subjects on their own. This is compulsory for university students to take part in registering for subjects. University students and elementary school students are different in the way of their registration subjects. University students are required to make their own decision about what subject they will take vice versa with the elementary school which is their timetable already prepared by the school clerk. Several people are needed in the process of course registration which is the academic staff who are managing the course registration, the lecturer who is responsible to teach each course offered, and the student themselves who registers for the course. Each person has the responsibility to complete the cycle of the course registration process.

II. BACKGROUND STUDY

The current system for course registration in UTM is quite complicated and takes much time to settle the course registration[1], [2]. First, the timetable schedule takes more than one draft to archive the final draft with no clash and redundant schedule. It is because the schedule is arranged by the staff manually [2]. That is why there are errors when arranging the schedule. Because of that, it makes students and lecturers keep on changing their timetable planning when the timetable draft keeps on changing as well. Second, there is no timetable generated when the user plans their timetable. It is hard for the user to identify if their course is clashing with another course or not. Third, there is two-time registration which is a "pre-registration session" and a "registration session". Pre-registration sessions are conducted for a student who is ready to make a registration without any problem regarding their previous course grade while the registration session is the final course registration session who are not able to complete their pre-registration session. It is much burden for students to make a course registration with double confirmation.

From the problems above, this project proposed a new functionality with an attractive design. First, the system can identify and avoid a clash timetable. With this functionality, there should only have one final draft of the timetable, and may be easier for the lecturer and the student to manage their timetable. Second, Students can view their timetable before making a course registration. It is easier for the student to manage their schedule and plan for their timetable. Third, the system shall make a one-time registration for all courses in

UTM. This function will make the process of course registration run smoothly and well-organized.

III. METHODOLOGY

The proposed CRS system is a web-based system. The web framework used for this project is the ReactJS framework. JavaScript knowledge is dominant in using ReactJS for implementation. While the proposed Web-based online application CRS is concerned about UI/UX implementation, ReactJS provide painless interactive UIs [3]. The efficiency of updating and rendering the system components give a benefit to this system since the system can be clarified as a big system. The backend framework of CRS that has been chosen is using the Loopback framework. NodeJS and TypeScript knowledge is dominant in using Loopback for implementation. Since the CRS is a web service application, REST API is the basic thing that needs to implement in handling API requests/responses. The loopback framework is a framework that provides an OpenAPI specification-driven REST API [4]. It is easiest for developers to test the backend connection by using the REST API that is already provided by the loopback framework.

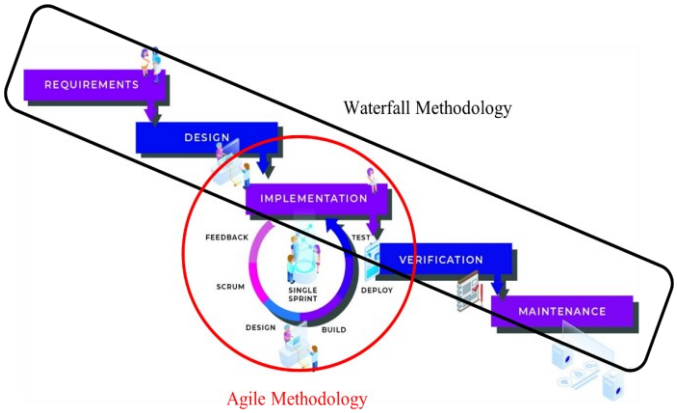

Fig. 1. CRS Methodology

Fig. 1 above shows a graphic explanation of the system development life cycle that is used for the CRS project which inspired by hybrid agile waterfall methodology [5], [6]. The fact that this hybrid is chosen is due it's systematical documentation while able to offer flexibility for development phase [6]–[8]. The methodology itself have gain its popularity in recent year due to the numbers of its utilization [5], [7], [9]. in There are 5 phases in waterfall methodology which are Requirement gathering, System design, System implementation, System verification, and system maintenance. The system implementation phase of the waterfall methodology will concentrate more on the agile methodology. There are 6 phases in agile methodology which are system feedback, scrum, design, build, deploy and test. Each phase is repeated according to the number of sprints.

A. Requirement Gathering

The technique used for gathering data is brainstorming with the user, observation from the previous registration process, and handling a survey. The users of the system are students, lecturers, and academic staff.

B. System Design

Designing a system architecture, defining module and subsystem, and designing the prototype and interface are conducted in this phase. System design can be categorized as a backbone of system development because this phase is the main information for starting a development process. Enterprise Architecture Software has been chosen to design the system architecture. All UML diagrams such as use case diagrams, component diagrams, activity diagrams, and sequence diagrams are designed using this tool.

C. System Implementation

System Implementation is the core phase of system development. This phase will be agile methodology since this methodology allows for changes and focused on delivering an agreed-to product [10]. There are 6 phases in this methodology, and it is repeated according to the number of the sprint backlog. Fig. 1 previously shows how the agile methodology is embedded into the waterfall methodology. The combination of these methodologies is called an Agile-waterfall or hybrid methodology.

The frameworks used for this CRS system are ReactJS for web development and Loopback for backend development. Both technologies are focusing on JavaScript and TypeScript language. By using these frameworks, the code architecture is manageable and structured. Fig. 2 shows the project design on how the code architecture is connected with the database service.

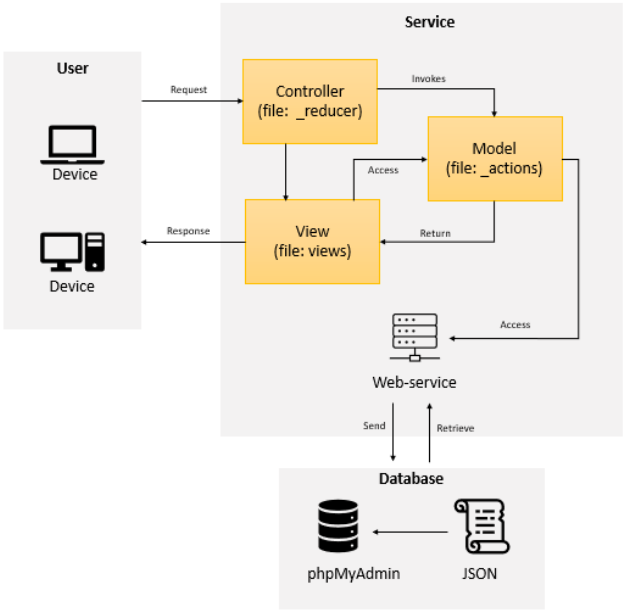

Fig. 2. CRS code Architecture

The proposed CRS is integrated with a database which contains information about students, lecturers, and courses. For security and integrity purpose, the development of the proposed CRS is only allowed to view the database to avoid migration data issues. The way to retrieve the data from the database is by JSON-based web service provided. The provided service is related to access to reusable e-learning data from the faculty of computing timetable web application. The data to be reused are classified and the data entities are further specific by their attributes. Fig. 3 is an example of the possible attributes to extract entities.

```
sesisemester -> entity=sesisemester

pensyarah -> entity=pensyarah&session_id=???&sesi=yyyy/yyyy&semester=[1|2]
subjek    -> entity=subjek&sesi=yyyy/yyyy&semester=[1|2]
pelajar   -> entity=pelajar&session_id=???&sesi=yyyy/yyyy&semester=[1|2]&limit=num_&offset=num_

pensyarah_subjek -> entity=pensyarah_subjek&no_pekerja=???
pelajar_subjek   -> entity=pelajar_subjek&no_matrik=???

subjek_seksyen   -> entity=subjek_seksyen&sesi=yyyy/yyyy&semester=[1|2]
subjek_pelajar   -> entity=subjek_pelajar&session_id=???&sesi=yyyy/yyyy&semester=[1|2]&kod_subjek=???&seksyen=???
subjek_pensyarah -> entity=subjek_pensyarah&kod_subjek=???[&sesi=yyyy/yyyy&semester=[1|2]&seksyen=???]

kurikulum               -> entity=kurikulum
kurikulum               -> entity=kurikulum&cohort=yyyy/yyyy
kurikulum_subjek        -> entity=kurikulum_subjek&id_kurikulum=???
kurikulum_subjek_elektif -> entity=kurikulum_subjek_elektif&id_kurikulum_subjek=???

ruang         -> entity=ruang&kod_fakulti=???&kod_ruang_like=???
jadual_ruang  -> entity=jadual_ruang&sesi=yyyy/yyyy&semester=[1|2]&kod_ruang=???
jadual_subjek -> entity=jadual_subjek&sesi=yyyy/yyyy&semester=[1|2]&kod_subjek=???&seksyen=???
```

Fig. 3. Example of JSON web-service

Even there is a bundle of data can be collected, a private database is also needed to perform a fully CRUD function. It is because the provided JSON Web service does not allow the system to manipulate the data. phpMyAdmin is a private DBMS of CRS.

D. System Testing

System development is developed by humans. That means there is a possibility to have some defects and failures in the system development life cycle [11], [12]. The testing approach is the way to reduce the possibility of defects and failure in system development [13]. This testing phase is important to ensure the quality of the system and can be the final wall of preventing system bugs before the system launch.

There are four phases of a testing area in the system development which are unit testing, integration testing, system testing, and acceptance testing. System testing is the third phase in the testing area of system development. In system testing phase will determine whether the system meets or does not meet user specifications after the system implementation phase.

E. Deployment

Deployment is the final phase of Software Development Life Cycle (SDLC) which is this phase is ready to be delivered and used by the real user. All documents and products will behand over and authorized by the stakeholder which is the academic office of the Faculty of Computing, UTM. A Standby unit will be formed for the backup system once the CRS is open until the end of the course registration process. It is to make sure the process of registration by using the proposed system is stable and to reassure the stakeholder since it is a new release system.

IV. USER REQUIREMNT AND DESIGN

Requirements analysis involves requirements analysis, project design, database design, and system interface design. The Integrated Modeling Language (UML) is used to provide a clearer picture of this Course Registration System Web-Based Application.

Use case diagram is used to present the relationship between the actor and the use cases. Fig. 4 shows the use case diagram for CRS. Based on the diagram, there are three main actors in the system, which are the student, lecturer, and academic staff.

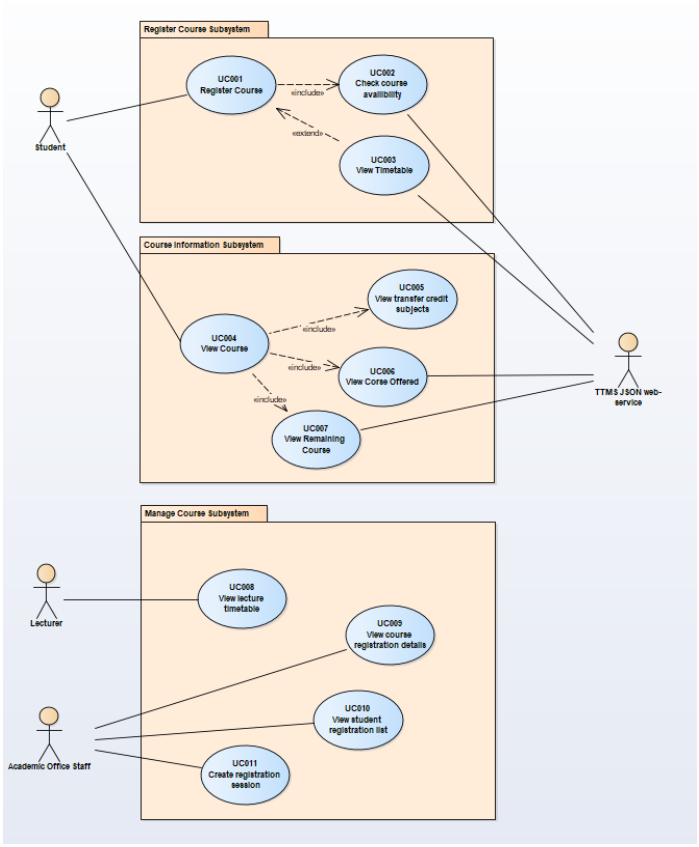

Fig. 4. CRS Use Case Diagram

From the diagram, there are twelve different use cases. Each use case describes various functional requirements of the system.

UC001	
Use case	Register Course
Actor	Student
Description	The student is able to register for a new course for the upcoming semester

UC002	
Use case	Check course availability
Actor	Student
Description	The student is able to view the course availability

UC003	
Use case	View timetable
Actor	Student
Description	The student is able to view the timetable for the registered course.

UC004	
Use case	View Course
Actor	Student
Description	The student able view the course

UC005	
Use case	View transfer credit subjects
Actor	Student
Description	The student able view transfer credit subjects

UC006	
Use case	View course offered
Actor	Student
Description	The student able to view the course offered

UC007	
Use case	View remaining course
Actor	Student
Description	The student able to view the remaining course

UC008	
Use case	View lecture timetable
Actor	Student
Description	The lecturer is able to view the timetable for the lecture course.

UC009	
Use case	View lecture timetable
Actor	Lecturer
Description	The staff is able to view the details of the offered course

UC010	
Use case	Register Course
Actor	Student
Description	The staff was able to view the list of students that completed their course registration for the new session

UC011	
Use case	Create registration session
Actor	Staff
Description	The staff able to schedule the course registration session

Based on these use cases, some of the graphical user interface (GUI) of the CRS are illustrated in Fig. 5 to Figs. 9. The figures are split into three types of users' main functions which are student, lecturer, and academic staff.

A. The Main Functions for Students

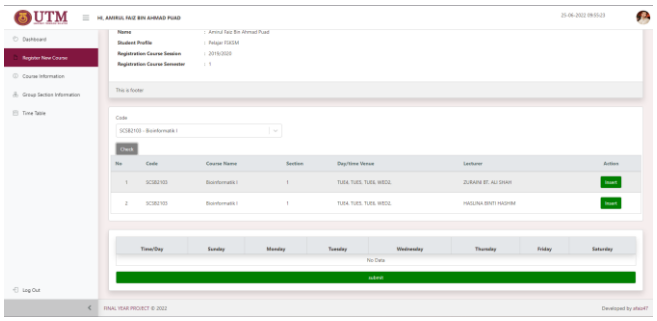

Fig. 5. Registration Process

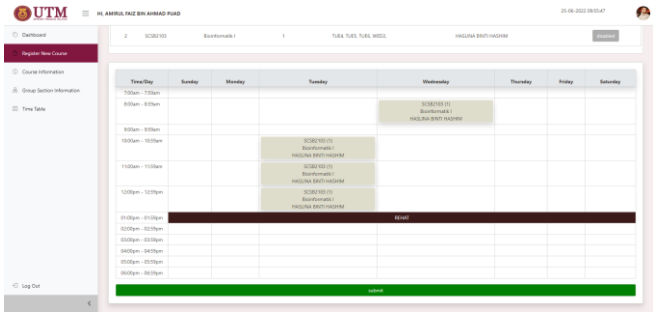

Fig. 6. Generate Timetable for Registration

B. The Main Functions for Lecturer

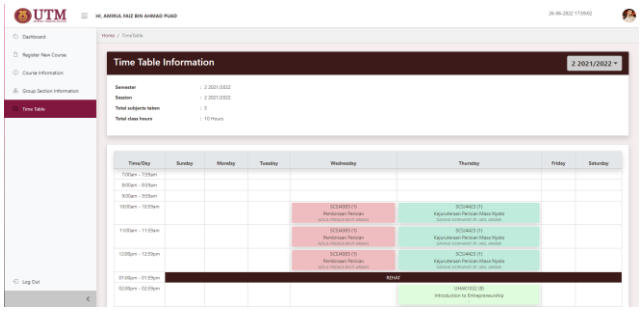

Fig. 7. View Previous Timetable Session

C. The Main Functions for Academic Staff

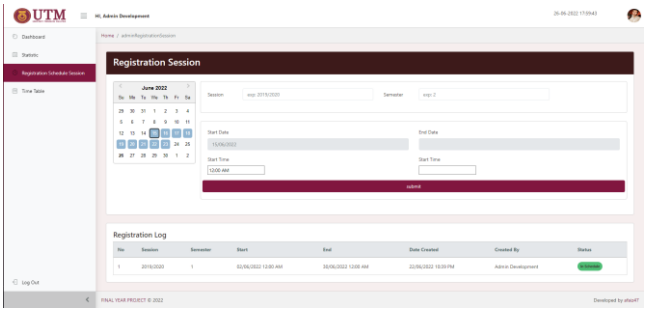

Fig. 8. Schedule Course Registration Session

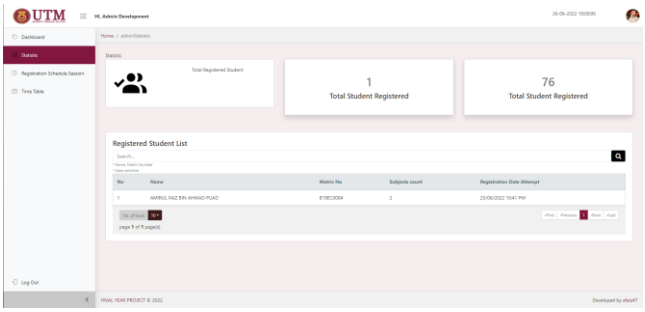

Fig. 9. View Statistics and Details of Course Registration

V. DISCUSSION

The purpose of this project is to develop a system that can help students to register for their courses sufficiently. Providing a visual timetable in the process of registration may give more benefits and provide user satisfaction when registering for the course. At the end of the development, the main purpose is to deliver a conducive way of course registration with put user satisfaction as the highest priority.

Testing has been made to ensure the system is delivered to meet user requirements and satisfaction. Blackbox and white-box testing have been made as test records and evidence for users. Several schools of computing students have been selected to conduct test cases. From the result, the system can be declared as a perfect system that encounters the current registration process and meets user satisfaction.

The application received positive feedback from all participants, indicating their satisfaction with it. Additionally, the majority of users noted that the application interface is user-friendly and straightforward, making it easy for them to complete their tasks. This positive feedback is encouraging, as a simpler and more user-friendly design tends to attract more users who want to delegate their tasks through the application. At the end of this project, CRS has been successfully developed to fulfill the requirements, objectives, and specifications that have been identified during the requirement gathering stage. The CRS system able to provide a complete feature that make students register for a course with less burden and less complexity through the visualized table feature.

VI. CONCLUSION

In conclusion, this project is challenging because it is a big project, that needs to beat the standard of the current system, and the system is developed completely by one person. ReactJS and Loopback framework used in developing this project is one of the right steps to implement because of the technology demand nowadays. The primary objective of this project is to make a student feel satisfied and secure in making the course registration process. Just adding on a simple visual timetable function may change the impact of the course registration process. Another new major function that is implemented in the current registration process is the student can see their members after making a registration. It is much more convenient for the student to get along with their course mate before starting the new semester.

For future improvement, some of the suggestions are to able the system to suggest the remaining course that able the student to register based on their study plan and prerequisite course. Besides, an admin can generate the list of registration into an excel format that easier for the academic staff to manage the student registration process to the UTM Academic Management Division.

ACKNOWLEDGMENT

Sincere gratitude to the esteemed lecturers and dedicated faculty staff for their invaluable support and guidance throughout the completion of this project. The unwavering commitment to fostering a conducive learning environment has been instrumental in this project success.

CONFLICTS OF INTEREST

The author(s) declare(s) that there is no conflict of interest regarding the publication of this paper.

REFERENCES

[1] N. A. H. M. Rodzi, M. S. Othman, and L. M. Yusuf. (2016). Significance of data integration and ETL in business intelligence framework for higher education. *Proc. - 2015 Int. Conf. Sci. Inf. Technol. Big Data Spectr. Futur. Inf. Econ. ICSITech 2015*. 181–186. Doi: 10.1109/ICSITECH.2015.7407800.

[2] H. Mammi, L. Y.-I. J. of I. Computing, and U. (2021). Timetable scheduling system using genetic algorithm for school of computing (tsuGA). *IJIC, 11*(2), 67-72. Doi: 10.11113/ijic.v11n2.342.

[3] K. Bernsmed, D. S. Cruzes, M. G. Jaatun, and M. Iovan. (2022). Adopting threat modelling in agile software development projects. *J. Syst. Softw.*, *183*, 111090. Doi: 10.1016/J.JSS.2021.111090.

[4] API Economy made easy with LoopBack 4. Doi: 10.5555/3172795.3172838.

[5] N. Prenner, C. Unger-Windeler, and K. Schneider. (2020). How are hybrid development approaches organized? A systematic literature review. *Proc. - 2020 IEEE/ACM Int. Conf. Softw. Syst. Process. ICSSP 2020*. *10*(20), 145-154. Doi: 10.1145/3379177.3388907.

[6] N. Yahya and S. S. Maidin. (2022). The waterfall model with agile scrum as the hybrid agile model for the software engineering team. *2022 10th Int. Conf. Cyber IT Serv. Manag. CITSM 2022*. Doi: 10.1109/CITSM56380.2022.9936036.

[7] A. Mishra and Y. I. Alzoubi. (2023). Structured software development versus agile software development: A comparative analysis. *Int. J. Syst. Assur. Eng. Manag.*, *14*(4), 1504-1522. Doi: 10.1007/S13198-023-01958-5/FIGURES/4.

[8] G. Theocharis, M. Kuhrmann, J. Münch, and P. Diebold. (2015). Is water-scrum-fall reality? On the use of agile and traditional development practices. *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*. 9459, 149-166. Doi: 10.1007/978-3-319-26844-6_11.

[9] M. A. Akbar, K. Smolander, S. Mahmood, and A. Alsanad. (2022). Toward successful DevSecOps in software

development organizations: A decision-making framework. *Inf. Softw. Technol.*, *147*, 106894. Doi: 10.1016/J.INFSOF.2022.106894.

[10] I. Mergel, S. Ganapati, and A. B. Whitford. (2021). Agile: A new way of governing. *Public Adm. Rev.*, *81*(1), 161-165. Doi: 10.1111/PUAR.13202.

[11] A. Akhtar, B. Bakhtawar, and S. Akhtar. (2022). Extreme programming vs scrum: a comparison of agile models. *Int. J. Technol. Innov. Manag.*, *2*(2). Doi: 10.54489/ijtim.v2i2.77.

[12] W. P. Neumann, S. Winkelhaus, E. H. Grosse, and C. H. Glock. (2021). Industry 4.0 and the human factor – A systems framework and analysis methodology for successful development. *Int. J. Prod. Econ.*, *233*, 107992. Doi: 10.1016/J.IJPE.2020.107992.

[13] M. Khatibsyarbini *et al.* (2021). Trend application of machine learning in test case prioritization: a review on techniques. *IEEE Access*, 166262-166282. Doi: 10.1109/ACCESS.2021.3135508.