Project Review

The Rotary Youth Driver Awareness (RYDA) application, an online quiz application, was proposed as a solution to the need of educational tool for the road safety awareness in young drivers. The project management and development of the enterprise application was managed and carried out by a team of three members. Each member in the progress of the project was involved extensively and contributed equally to the deliverables of the project. The dedication and commitment from each team member lead the project to success through each project phases. Further review, is summarized and evaluated on each development stage, as follows.

Project proposal and planning

The proposal for the project encompassed an enterprise application with adherence to the three-tier software architecture. During the proposal phase, it was clear that we were developing an enterprise web application that would allow young students to take a quiz on road safety awareness. But the toughest decision was to select a programming language for the development of the application. Even though, through team discussion and members level of competency, we decided to implement Java Enterprise Edition (J2EE) architecture, with Java Server Faces for presentation layer, Managed Beans and Enterprise Java Beans (EJBs) for the business logic, and Java Persistence API (JPA) for persistence layer.

The major constraint for the project was the time, as we had the estimated time frame of 65 days, we planned the project development through a complete work breakdown structure, and defined the milestones based on the course assessment schedule. The quality matrices and risks were identified and quality control and risk mitigation was planned as early in the planning phase, to develop reliable and secure application while minimizing the project risks.

Requirement specification

The project proposal and planning phase eased the process of requirement identification, as proposal was based on the need of an educational tool for RYDA program for road safety awareness. The functional requirements for the online quiz application were identified through analysis of the proposal and need request of the RYDA program. Further, we identified non-functional requirement to enhance the interactivity of the user, enable security within the application, ease the debugging and further development of the application, and endorse reliable and flexible application. Further, the requirements for the application were prioritized according to the importance to the functionality of the application and the use cases of were created to map them with the requirements.

Design

During the design phase of the project, software architecture was design to meet the need for the proposed software application. The application was modelled into three layers of Model, View and Controller. The Model layer comprised all the business model and business logics to implement the application’s entities, JPAs, EJBs and backing beans, while view layer included JSFs to handle the presentation for the application and the controller comprised J2EE Faces Servlets to handle the request and responses from the users. Further, the classes, user interfaces and behavioral for the application were modelled to map the identified requirements. During the modelling of classes, user interfaces and behaviour, the time constraint for the project was considered extensively, as inclusion of extra functionality and features for the application could impact the schedule the project completion. So, minding the risk, we designed the classes, user interfaces and behaviour for the system limiting to the identified requirements. Until the design phase, we didn’t encounter any risks and issues for the project management.

Implementation and testing

During the implementation of the project, the application was implemented as per the design specification except for the minor changes or fixes. The implementation of database design to the application triggered separation of student and administrator as separate entities. Along with, addition of classes was important for improvement of application functionality. Apart of these minor changes, the identified requirements and design specification were implemented progressively in the project.

The implementation phase was completed within the planned schedule and was effectively executed with regular communication between the team members. The software development and version controlling for the project was managed using the GitHub repository, which assisted in team collaboration, code review and issue tracking. Further, bugs and errors identified in the software code were effectively resolved through debugging and reviews amongst the team members.

Thus, through teamwork, effective communication, efficient project management techniques and dedication of team members, the project was successfully completed within the estimated schedule and quality performances.

Lessons Learned

Applied Software Engineering process

We were successful in executing the software engineering process throughout the project. It started with analyzing the requirement, documenting the proposed design, designing an application, coding and debugging, testing an application. We learned to document on every step of process including the progress of the project. We quickly realized that the document would be the roadmap of our project.

Project management skills

The development time frame for our project was 65 days and milestones were divided into small milestones/task which were scheduled using the Microsoft Project 2016. With the group of 3, we kept on changing the Project Manager where everyone was PM once. The Project Manager was responsible for planning the project, leading and controlling the project. To ensure this, Project Manager used to call for the meeting every day, take an update then assign the task for the rest of the day. By this, we gained some practical knowledge on how the Project Manager should work in a real-time environment.

Leadership skills

We learned some leadership skill while we were a Project Manager. We tracked the progress of the project every day, and PM would ask us to either continue the task or assign developer a new task. PM documented the progress of every developer.

Team work

We learned to work as a team for this project. We coordinated with each other on every aspect of the project. Some days we worked together and some days we worked separately. Even though we worked separately, we were communicating with each other thru phone, skype. If anyone of us had any issue during development, we shared the issue in group and found solution for each other. We used Git for version control and Github to store our project in a repository where all our completed work could be shared in the project.

Time management

The project had a time constraint, where we had to propose, plan, develop and deliver the project in 65 days. In the allocated time frame, we acted smartly as we generated the basic function provided by the Netbeans (IDE) and developed the required functions and tried using the same code wherever possible rather than wasting our time in duplicating the code.

Risk management

As we moved on to develop the project, we were aware of possible risk for the project. Since, we were in a team, we learned to communicate any risk possible and stand by together on any risk occurred and tackle the situation. We had a risk of “overloaded by other course”, as we had assignment pressure from another course. During those period, some of us would be taking over the other’s work and we were able to eliminate the risk.

Quality management

Quality of the project was our major concern. The project should not only function well but also to look good as it should be user friendly. We were successful to develop a responsive site and user-friendly navigation buttons and links in the front-end. Whereas in back-end, we had flexibility to create question from the question menu and from the quiz menu. The student report could be visible from student list as well as from the reports menu. Finally, all the functionality was documented, user manual for Admin and Users are of high quality which addresses every function explained step by step as well as labelled in the screenshot of every page.