

Develop a Scalable and Serverless Client-based Application using Agile Methodology

G.A. Dhanush^a, Divya Rathi^b, G. Harini^c, T.P.V. Krishna Teja^d, Priyanka Kumar^{e*}

^aComputer Science and Engineering, Amrita Vishwa Vidyapeetham, Coimbatore, India.

Email: cb.en.u4cse17415@cb.students.amrita.edu

^bComputer Science and Engineering, Amrita Vishwa Vidyapeetham, Coimbatore, India.

Email: cb.en.u4cse17416@cb.students.amrita.edu

^cComputer Science and Engineering, Amrita Vishwa Vidyapeetham, Coimbatore, India.

Email: cb.en.u4cse17541@cb.students.amrita.edu

^dComputer Science and Engineering, Amrita Vishwa Vidyapeetham, Coimbatore, India.

Email: cb.en.u4cse17465@cb.students.amrita.edu

^{e*}Computer Science and Engineering, Amrita Vishwa Vidyapeetham, Coimbatore, India.

E-mail: k_priyanka@cb.amrita.edu

Article History: Received: 11 January 2021; Revised: 12 February 2021; Accepted: 27 March 2021; Published online: 10 May 2021

Abstract: Software Development Lifecycle plays a pivotal role in developing large-scale applications with numerous features by a large team of people in a fixed timeline. There are a variety of software development models available. One such popular model is called the Agile software development model. This is a popular framework used by a large number of companies today and has proven successful in developing small-scale to large-scale applications. In this paper, we gather the results of developing an application named course review portal that can be used to gain insights on courses available in universities. This will be beneficial for both the institution and the students in providing better elective choices and choosing the optimum choice respectively. The entire application runs client-side and communicates with a cloud database to fetch data. Continuous Integration and deployment are performed with Gitlab's CI/CD tools. We further discuss the use of Scrum framework of agile methodology and present the effectiveness of the framework used and the novelty of the application built.

Keywords: Agile Method, Scrum, Serverless Application, CI Pipeline.

1. Introduction

Software Development Life Cycle describes the lifecycle of development of an application, from its inception to the end of development and maintenance of the software. Following a structured approach to the development of software makes it easier for large teams to work on complex projects that involve application development [11]. There are a variety of methodologies that have been introduced to describe the life cycle of application development that have different properties. The methods range from agile to iterative to sequential development of an application. Each methodology has its own set of advantages and disadvantages [1].

The software model that has been used to develop this project is called the agile method. This methodology enables to dynamically plan features, rapidly prototype them and work in cross-functional teams to quickly build applications. This model is subdivided into different frameworks like Scrum, XP (Extreme Programming), rapid application development (RAD) etc. For this application, the Scrum framework was used to plan and iterate through the development cycles. It is specifically designed for team sizes of ten or fewer members. This framework defines sprint cycles in which features of the application are rapidly developed for a span of two weeks up to a month depending on the workload promised to be completed for that sprint.

The application that we have developed can be used to gain information about a variety of courses available in the universities and allow students to provide a review for these courses. This application is a web-based application built using a web framework called Nuxt.js. The application runs entirely on client-side and communicates with the database and is hosted on the cloud [12]-[14]. While developing the application, it was made sure that every module had a separate test case. This application was linked to Gitlab's continuous integration tools so that the changes made to codebase triggers actions that build the application and automatically deploy if all the stages of continuous integration pipelines (Refer Section 3) succeed.

Road map: The paper is divided into the following sections. The merits of agile methodology and Scrum Framework, and the choice to develop an application to review college courses is discussed in Section 2. Then, the description of project-specific details and the technologies used are described in Section 3. In Section 4, the implementation details and the results obtained in developing the application is discussed.

Finally, a conclusion is derived on the effectiveness of agile methodology, the novelty of our application and future plans to extend our application.

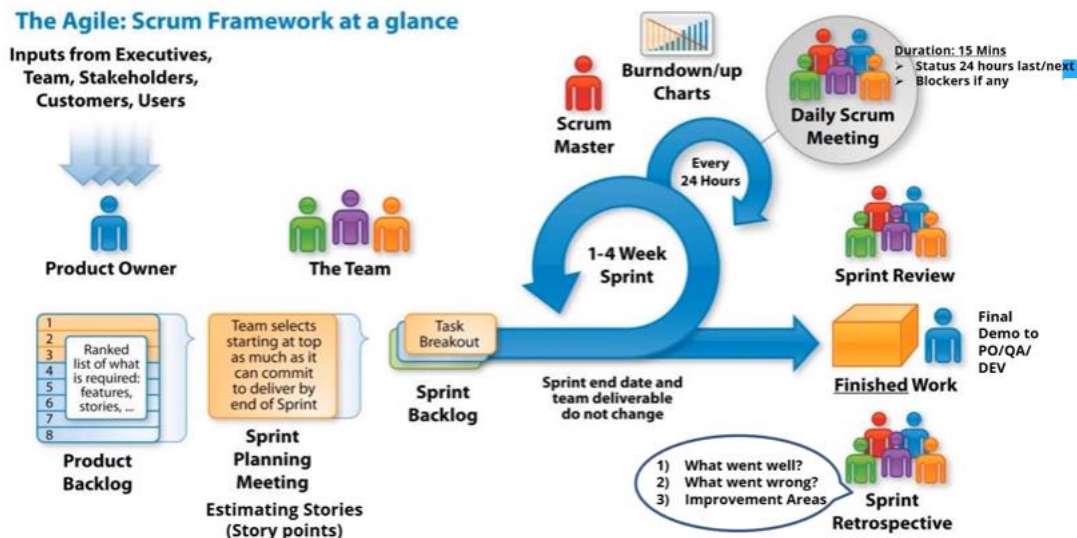


Fig. 1. Software Development Lifecycle in Agile Process

2. Background Study

Companies have been adapting to agile methodology from traditional waterfall model for software development. 75% of companies, including IBM, Google and Spotify, who have adopted the same stated that their major reason for this adoption being faster software delivery and many such reasons [3], [4]. So Agile methodology is a growing choice of companies in software development due to its benefits. Hence, we chose the same methodology for our software development. All universities offer courses to students. Core courses are mandatory for the students to complete their degree and elective courses are too available that help diversify their knowledge.

Similarly, we have explored that other colleges provide course registration portal for students to choose their elective from. In IIT Kanpur, elective courses are available for their college graduates thus increasing the flexibility for students by providing 12-12.4% core electives, a 3-5.7% open electives and 4.3-6% department electives [5]. This elective choice is made through a course registration portal.

It is evident that these registration portals alone are not sufficient for students to pick their courses. This is especially true for electives because this is where a student is offered a choice to pick his preferred course. For one to pick a course, he/she has to receive feedback from his/her seniors to make their decision. This alone is not reliable and cannot present facts about the courses. Moreover, electives are becoming increasingly important and might even help in one's career path [6].

Hence, a course review portal was built where one can add experience about the courses they attended and also present with actual facts about all the courses that the university offers. It is seen how online shopping platforms function and would like to provide a review system with similar features. A report reveals that Online reviews impact purchasing decisions for over 93% of customers [8]. With this motivation, the application is modelled to have similar functionality as an online review portal. There are additional features like students can ask questions about the courses and faculties to reply for the same. With these features, the course review portal is an application where university students could get information about courses present in their college and make the right choices when they pick their elective courses.

3. Project Description

A. Software Development Lifecycle

Let us understand the Scrum framework in this section. Scrum is an agile process for developing complex products, while productively delivering products and services that truly delight customers. Scrum helps in

innovating faster, moving our ideas to delivery quickly driving to higher customer satisfaction. The requirements for the product from the clients are gathered and a product backlog is designed consisting of an ordered list of features. The features are then grouped into epics which are high-level descriptions of what clients want. For our application, we categorized the features into six epics namely Account Management, View Q&A/Review Section, Post Q&A/Review Section, Email Portal/Comparison section, User Experience and Miscellaneous Features. The epics consist of user stories. Each story explains a single feature that is requested by clients.

S No	Stories	Available Story Points	Req Story Points	Tasks	Hours
1	Creation of landing page	3	3	Divide the page into regions and add flexbox according to the wireframe	3
				Find design elements for different components	3
				Create card and list component for the webpage	2
				Create a search field for the webpage	1
				Perform code analysis and unit testing with SonarQube and Jasmine	2
				Test the built webpage with Selenium tool	1
2	Creation of page for account management	2	2	Add page to view account details	3
				Creating a database	3
				Add edit button	1
				Setting up Jasmine for our project	1
3	Creation of Login Page	3	3	Creating a database	3.5
				Creation of login form	3.5
				Create submit button	1
				Write query for validation of user account	3
				Setting up Selenium for our project	1
				Creation of list format in the displaying page	2
4	Creation of page for displaying search results	2	2	Providing fuzzy search functionality	3
				Adding card elements to the list view that redirects to the corresponding course page	2
				Setting up SonarQube for our project	1
				Add page for Displaying Course	3.5
5	Creation of page for course description	3	3	Fetch details about the course from database	2
				Fetch and show the reviews	2

Fig. 2. Division of stories into tasks

We first gathered features that are required for our application and categorized them into one among the six epics. Once these stories are known, these are broken down further into subtasks. Subtasks are a set of concrete tasks that need to be completed to implement a story (feature). Following is the part of the set of tasks for the stories we have listed.

Time taken for each task is noted and accumulated to story points. Each story point refers to 4 hours of work. Accordingly, story points estimate the duration of completion of a story. Once this categorization is done, developers get ready to prepare for the sprint cycle. The project makes progress in a series of sprints. Each sprint would typically be of 2-4 weeks. During a sprint, the product is designed, coded and tested. Each sprint starts with sprint planning that establishes a sprint goal and duration and prioritizes features from the product backlog forming a sprint backlog.

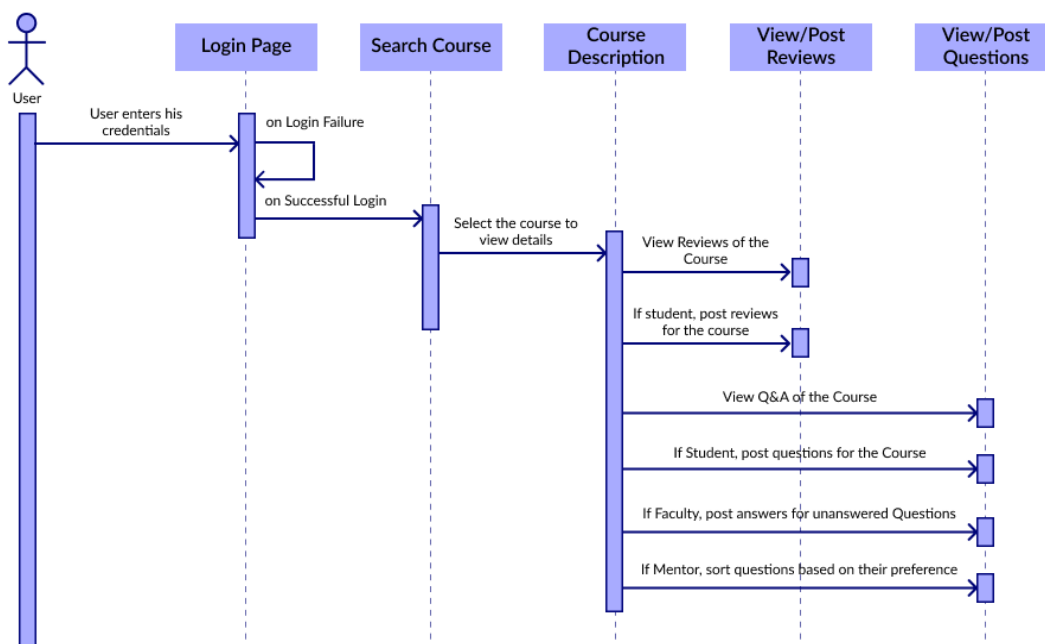


Fig. 3. Sequence Diagram of Course Review Portal

Each day in a sprint, the team conducts a scrum meeting where developers discuss their progress. At the end of the sprint, two ceremonies are held by the team: sprint review and sprint retrospective. The sprint review is where the team reviews if the work has been completed as planned at the beginning of the sprint. In the sprint retrospective, the team looks into the past sprint and figures out dos and don'ts and what to be continued. This cycle repeats iteratively until the end of the development lifecycle of an application.

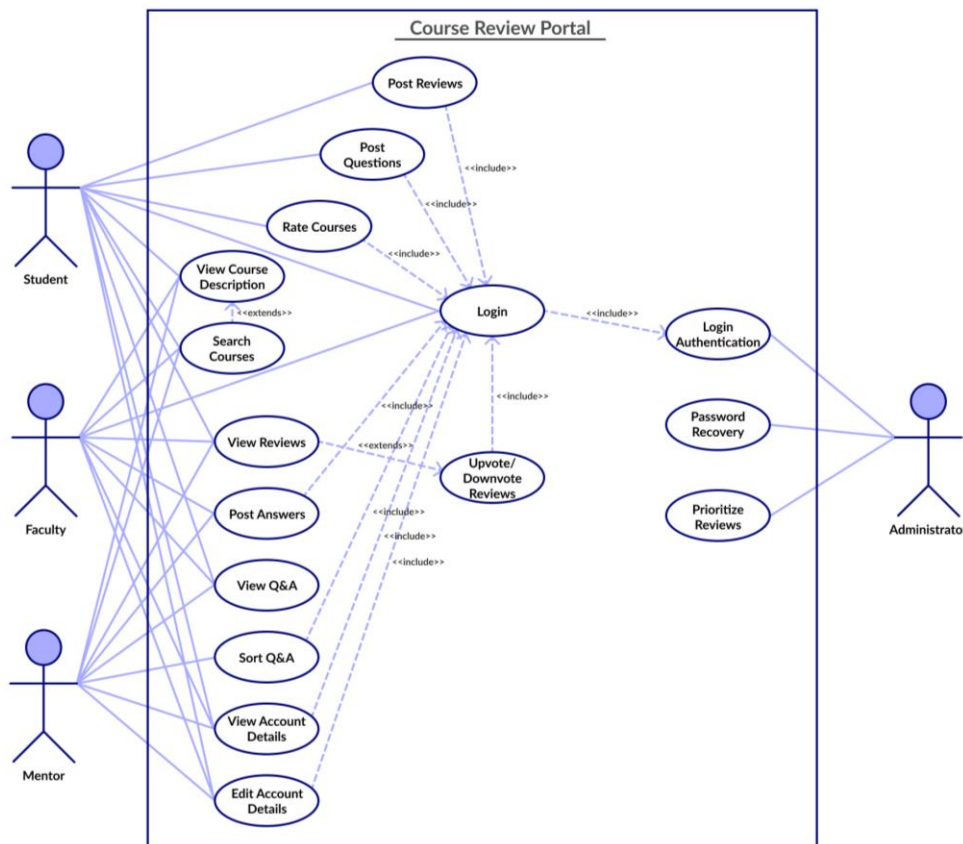


Fig. 4. Use Case Diagram of Course Review Portal

For our application, we worked in a set of three sprint cycles. In the first sprint cycle, we prepared product backlog, organized stories into epics and planned sprint backlog for the upcoming sprint cycle. In the second sprint, we had implemented a minimum viable product of our application. In the third sprint cycle, we wrote integration test cases and worked with setting up continuous integration for our application using Gitlab CI/CD tools. Apart from this, we made two different types of UML diagrams for our application, namely use case diagram and sequence diagram. Use case diagram is seen in Figure. 4 and describes all the use cases of our application and sequence diagram is seen in Figure. 3 and it describes the flow of our application.

B. Tools and Technologies Used

Jira [9] is a powerful work management tool which is used by most of the companies to track the progress of each of their teams. Initially, it was designed as a bug and issue tracker, but as time progressed it became very helpful in work management. For teams who are building using agile methodology, Jira offers scrum and Kanban boards for visualization. Time tracking capabilities and real-time performance analysis help organizations to closely monitor their teams. It also provides burn-down charts that graph the number of features that remain to be developed.

Nuxt.js is one amongst the popular JavaScript frameworks that ease the process of building web applications. It is a higher-level framework that is built on top of another framework called Vue.js. Nuxt.js abstracts away the details of server and client code distribution so you can focus on application development. It internally creates a connect instance that we can add our own custom middleware to. A middleware provides server-like functionality to a completely client-based application. To communicate with a cloud database and to manage the application state, using a middleware makes the process easier. Also, sharing data between pages of the application is also handled by this middleware. For the frontend, we used **Bulma**, a CSS library to style the components present in our application. This completes the set of tools required to build the client-side visual interface.

Firestore is a Backend-as-a-Service (BaaS) that provides functionalities that can help with the backend development of any Android, iOS or web application. We don't need to manage servers or write APIs. Firestore is the server, the API and the datastore, all written so generically that we can modify it to suit most needs. Firestore auth has a built-in email/password authentication system. It integrates directly into Firestore Database, so we can use it to control access to the data.

Gitlab CI/CD [10] is a powerful tool built on GitLab that allows us to apply continuous methods such as continuous integration and continuous deployment to our software with no third-party application or integration needed. Gitlab CI/CD not only executes the jobs which are set but also shows what's happening during execution just like a terminal. Once the strategy is created for the application, GitLab runs the pipeline. Pipelines are the top-level component of CI and CD. The pipeline comprises Jobs, which define what to do and Stages, which define when to do runners execute jobs. If all jobs in a stage succeed, the pipeline moves to the next stage. We can define any number of stages in this pipeline and each stage performs a specific task on the application. Primarily, the stages might be divided into setup, test, build and deploy. If all these stages succeed, the newer built version will get hosted on the cloud, or else, the previous successful build will remain to be hosted.

Testing Tools - For this application, we performed unit tests, integration tests and UI tests. Unit tests and integration tests can be written with a test framework known as Jest. It is a modern unit testing framework that works with popular JavaScript frameworks like React, Vue.js, Angular etc. Any JavaScript application can be tested with the Jest framework. For writing UI tests, a tool called Selenium was used. It is an automated testing framework to validate the UI of the application across various web browsers. Selenium scripts can load the browser, navigate to our application and perform a variety of UI tests on the application. Finally, another tool called SonarQube was used for static analysis of code to detect bugs, code smells and security vulnerabilities. With all these testing tools, we were able to ensure the quality of our code.

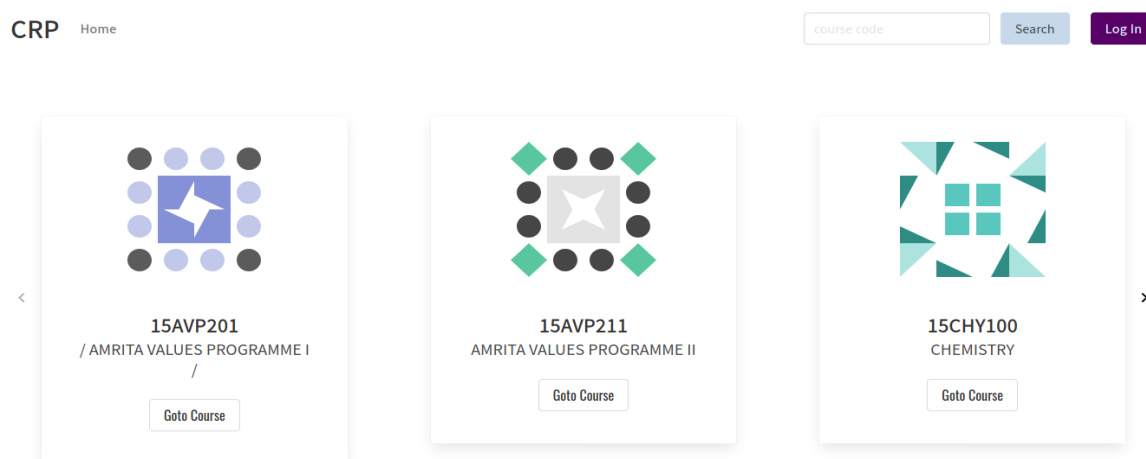


Fig. 5. Frontend of our application

4. Implementation and Results

C. Application-specific Details

Beginning with the frontend of our application, we had to set up Nuxt with Node's npx package management utility. It bootstraps the process of building our application by providing a starter template. Bulma, the CSS framework, also gets installed along with this. A Nuxt config file is configured with the necessary modules, plugins and middleware authentication is set up in the router. In the middleware, we used cookies to store the login details and state of the user. To manage the state of the application, we have used Vuex store (a state container that manages the entire state of the application efficiently). It uses actions and mutations to make changes in the application predictable. We extensively utilized the Vuex store to write the application logic that communicates with the firestore database. To perform UI testing, we wrote the Selenium test script in Python.

To use Gitlab CI/CD, the first step was hosting the application codebase in a git repository and specifying the different stages of the build pipeline in a file called `.gitlab-ci.yml`, located in the root path of the repository. The pipeline for our application consists of four stages: setup, test, build and deploy. The setup stage includes installing the libraries, packages and dependencies used. The test stage executes two scripts, one does unit testing of the

application and the other does UI testing on the application. The build stage builds the application, and the deploy stage deploys the application on Netlify, a cloud hosting platform on which we have deployed our web application.

D. Application flow

The user of the application is initially presented with a screen that displays a list of courses in a sliding carousel. This makes the user feel the application intuitive to use as he/she can begin to scroll the pages to view more courses and he/she can access all the courses from the home page. There is a navbar (Navigation Bar) present on the top of the page that guides the user across the different pages of the application. It contains a search bar that can be used to search for different courses. Besides the search bar, a login/logout button is present. Apart from this, the navbar holds various pages accessible by the user.

When the user clicks the login button, he/she is redirected to the login page. Here, the user can enter his credentials and login to the portal. These credentials are pre-registered and will be provided by the college to each individual student/faculty. This is to prevent fraudulent registrations in the application. A user can have a student, faculty or a mentor privilege. If a user is a student, he/she is allowed to post reviews and ask questions for courses. If a user is a faculty member, he/she is allowed to answer questions. If a user is a mentor, he/she is allowed to answer questions and also sort the questions according to priority. Once the user logs in, an account section gets visible in the navbar. This page contains user details. In a future version, we would like to add courses registered by students and courses handled by faculty/mentor in this same interface.

The user can click on “view” under the course card for a course on the homepage and is redirected to a new page to view more details about the course. This page is accessible by both registered and unregistered users. This page displays course details, syllabus and ratings. At the bottom of this page, two buttons that redirect to the reviews page and questions page respectively are present. The reviews page contains the textual reviews of that particular course. If the user is registered and is a student, he/she can post reviews from the same page. The post questions page contains questions and its corresponding answers organized neatly on the page. According to the privilege level, the user can perform actions on the page as mentioned in the previous paragraph. With this, we have described how users can interact with our application.

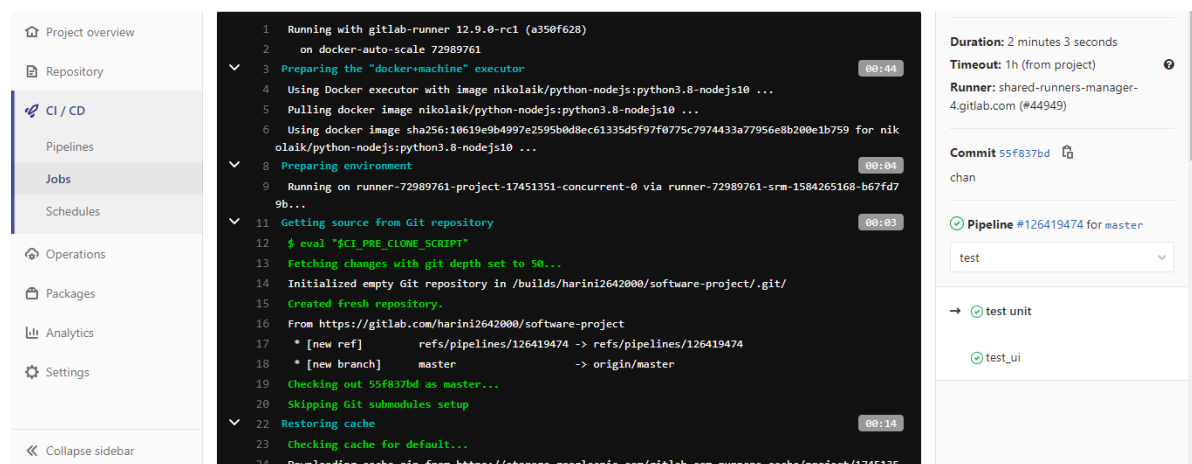


Fig. 6. A running process of GitLab Continuous Integration

E. Results

Scrum framework greatly boosted our development productivity. We assigned priorities to different tasks that we listed and assigned a set of tasks to each developer. Tasks were allocated to sets in such a way that one set of tasks does not depend on another set. This way, we were quickly able to develop application features. We made sure that we worked on the tasks for the specified time as mentioned in the sprint backlog. Also, daily meetings ensured that we were on the right track. SonarQube was used to monitor the quality of the code throughout the application development process. Following is the screenshot of the final SonarQube’s analysis.

Moving on, let us look at the advantages of our application being client-based and not necessarily relying on a server to function. Being client based means all the application logic happens on client-side, so there is no latency when data is processed or an event is triggered in the application. Also, the application can be loaded and cached in the browser’s local storage. This would significantly improve the load times of the application. Using Nuxt.js enables us to perform this. The data that is to be fetched from the cloud can also be cached so that the application

would fetch data only if there is any change in the original data. With all these features, it makes a web application to perform similar to a desktop application.

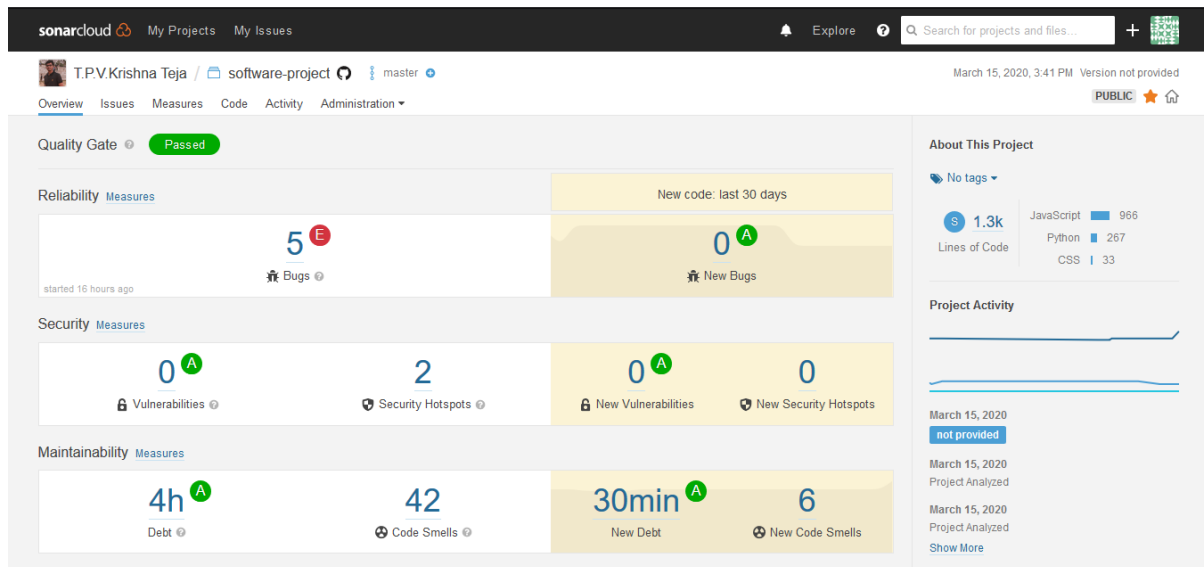


Fig. 7. Static Analysis performed using SonarQube

Finally, we would like to brief you about the continuous integration setup. As mentioned earlier, the CI pipeline is divided into four stages. In each stage, a specific set of tasks is performed. To perform these tasks, an environment must be built for the application to include necessary library packages. The build configuration of this environment can be specified with the help of Docker compose (Docker is a tool to containerize applications and docker-compose can be used to specify build configurations). We also built a separate docker image for UI testing as we could not find any prebuilt image to suit our requirement. Usage of docker makes it flexible to define the processes in the integration pipeline and these pipelines will be triggered to run when there is a change in the codebase.

Using all these tools, we successfully developed our application and tested it. Figure. 7 shows our applications' code quality. Out of 1.3K lines of code written, there were only 5 minor bugs and they were later fixed. The continuous integration pipeline ran tight test cases against our application and the test cases passed successfully. This quality code was deployed in the Netlify's cloud server. These results show that we were able to build a quality application in a span of three months and received positive results.

5. Conclusion

This application has been developed by a group of five undergraduate students and successfully implemented and tested as a part of the project-based Software Engineering undergraduate course. With the experience gained in this project, we plan to build more robust and scalable applications in the future. Finally, we would like to extend our current application by making changes to the user interface and deal with the security aspects of our application.

References

1. SDLC Models. [online] Available at: <https://existek.com/blog/sdlc-models/> [Accessed 10 February 2019].
2. Kanban Board. [online] Available at: <https://www.atlassian.com/agile/kanban/boards> [Accessed 10 February 2019].
3. Adopting Agile: The Latest Reports About The Popular Mindset. [online] Available at: <https://adevait.com/blog/remote-work/adopting-agile-the-latest-reports-about-the-popular-mindset/> [Accessed 10 February 2019].
4. How Agile Scrum Training Transformed These 5 Companies. [online] Available at: <https://www.quickstart.com/blog/how-agile-scrum-training-transformed-these-5-companies/> [Accessed 14 February 2019].
5. IIT Kanpur Online Portal. [online] Available at: <https://www.iitk.ac.in/> [Accessed 21 February 2019].
6. Movchan, Larysa & Zarishniak, Inna. (2017). The Role of Elective Courses in Students' Professional Development: Foreign Experience. *Comparative Professional Pedagogy*. 7. 10.1515/rpp-2017-0018.

7. Electives - IIT Madras. [online] Available at: <https://timesofindia.indiatimes.com/home/education/news/From-next-year-IIT-Madras-students-get-more-electives-to-pick-from/articleshow/45076712.cms> [Accessed 10 February 2019].
8. Online reviews impact purchasing decisions. [online] Available at: <https://www.thedrum.com/news/2017/03/27/online-reviews-impact-purchasing-decisions-over-93-consumers-report-suggests> [Accessed 14 February 2019].
9. *report-suggests* [Accessed 14 February 2019].
10. Jira. [online] Available at: <https://www.atlassian.com/software/jira> [Accessed 21 February 2019].
11. Gitlab CI/CD. [online] Available at: <https://docs.gitlab.com/ee/ci/> [Accessed 21 February 2019].
12. B.N. Prashanth, R. Venkataram, Development of Modular Integration Framework between PLM and ERP Systems, *Materials Today: Proceedings*, Volume 4, Issue 2, Part A, 2017, Pages 2269-2278, ISSN 2214-7853, <https://doi.org/10.1016/j.matpr.2017.02.075>.
13. Ambikadevi, Pramod, V. R., and Dr. Radhika N., "Cloud computing- A challenge paradigm in information Technology", *Journal of organisational management*, vol. VolX XV111, no. 4, 2013.
14. K. Divya and S. Jeyalatha, "Key technologies in cloud computing," 2012 *International Conference on Cloud Computing Technologies, Applications and Management (ICCCTAM)*, Dubai, United Arab Emirates, 2012, pp. 196-199, doi: 10.1109/ICCCTAM.2012.6488097. Ambikadevi, Pramod, V.R., and Dr. Radhika N., "Modeling the interrelationship between the enablers of cloud computing", *International Journal of computer Applications*, vol. 67, no. 20, 2013.