**Team <TEAM\_NAME>**

**Team Member 1** (email)

Team Member 2 (email)

Team Member 3 (email)

Team Member 4 (email)

Use a splash page image here [optional]

Use *LaTeX* if you wish, but use the general spacing and font/style you find here (1.5 spacing, 12 point font for text, etc.).

Be sure to submit a PDF (not a .DOC file) as your report. Overall it should be **15 to 20 pages**, including diagrams.

**1. Introduction**

Introduce your vision of the project and what you set out to achieve. What did you see as the core components and what were ancillary or bonus extras? If your view of the project was novel, or if it differed from that of your classmates in significant ways, then here is the place to say that and grab our attention.

**2. System Features**

What features did you implement and why? What features did you deliberately *not* implement? What features did you plan to implement but not actually do in the end? What features do you believe distinguish *your* project from the projects of your classmates.

The primary features include the ability to

**3. Project Design**

Tell us about the design of your system. What model does it follow? Does it use a web framework or is it launched as a standalone application from the desktop or the console? What software patterns did you use in its design? What third party libraries or components did you build upon? What aspects are you most proud of? What aspects would benefit from further development, or even refactoring? Feel free to use software diagrams here to provide a schematic overview of the project’s design, but only if they offer value for the space they take up.

Scentaur is a web based application that analyses user's code (Java) for code smells. It enables the user to upload their entire code base as a zip file which will then be analysed by the system. Scentaur guarantees easy accessibility, as it is a web-based application that solely requires a standard web browser. Unlike standalone-apps, the only requirement for a user is a web browser and a connection to the internet to access the application. The application runs java on the backend and uses HTML5, CSS3 and Javascript on the front end. We use Java Servlets using .jsp files to execute java code in the backend while at the same time providing user with their HTML, CSS and Javascript code to the browser.

For the front end, we made use of various JavaScript libraries to perform various functions. We used ajax to lazy load data in order to make the web page seem smoother to the front end user. We utilized highlight.js (<https://highlightjs.org/>), a JavaScript based syntax highlighting tool which we used to display the java source codes for each file. We also utilized (highlightjs-line-numbers.js) in order to insert line numbers when displaying the code onto the user's browser. The design of the web page was kept minimal and dark in order to better suit developers. We wanted the majority of the page to be concentrated on what matters rather than random objects and shadows.

In terms of the backend, we were planning on doing spring boot as an MVC. While we did make a slight progress with Spring Boot, we decided that it would be easier and much simpler if we just stuck to JSP and Java Servlets. In terms of libraries, [JAVAPARSER, TOMCAT, ..?]

We have used gradle to allow easy imports of libraries. But, we weren't able to find all the things we had to import within their repository so somethings had to be imported manually. Ideally we would have been able to import everything using just gradle.

In terms of improvements

[SOFTWARE PATTERNS?!?!?!?]

***What aspects are you most proud of?***

The biggest aspect of the project that we are proud of is our ability to make the project into a web app instead of a standalone jar file which would have been a much simpler way to achieve the goal. Turning the project into a web app not only increased the usability and accessability to users now and potential future users if we do plan on releasing this to the public, it also helped us learn a lot of about developing for the web using Java. While members on our team have had opportunities to work on the backend with languages such as Ruby and PHP, using Java to create the backend was a new challenge and we are happy to have taken that challenge.

***What aspects would benefit from further development, or even refactoring?***

We would have loved for the website to have more functionalities such as storage of reports for future reference, uploading of multiple projects at once. On the backend side, with more work a lot of the smells that have been listed as "Possible Code Smell" could have been moved up to "Likely Code Smell" using more constraints and checks.

**4. Successes and Failures**

Working as a group is a learning process; it is an explicit learning outcome of this course. So tell us what you learned from working in a team with your fellow team members. What problems arose, and how did you deal with them? How would you do things differently in the future, or if you had to do it all over again. Do not be afraid to discuss failures too: you will not be penalized for being open about the downs as well as the ups of your project.

Who did what in the end? Who deserves special mention for going above and beyond the call of duty? Who did less, or nothing at all? If you feel that a team member did not contribute, you may say so here, but do give that member the right of response within the report.

**5. Team Communication**

Team communication in general went much better than expected. Meetings were regularly held and plans were made. We had group meetings and calls on platforms such as discord and messenger where we discussed our plans and ideas, and critiques were well welcomed. These were crucial in ensuring that the team works as a unit instead of each person doing their own thing. Each person was assigned specific tasks for them to do before the next meeting. For our team, this method worked very well.

However, like all other teams, we have had our off weeks where due to the accumulation of work and other engagements, we were not able to do as well as we should have had. In terms of 'tools of software development', we were able to make proper use of a few softwares including Slack, Github, Discord and Messenger.

Slack was used as an official mode for communication and a place to store ideas, suggestions and report various issues. We created various channels for various purposes so that everything wasn't bundled up into one single channel making it difficult for us in the future to find information we would be looking for. We integrated github onto each of our slack apps so that we received constant updates and notifications for all changes that occurred on github. This was not limited to commits but also included various other things such as changes on the project board and issues raised. We were able to use @channel and @here to send notifications to all members to raise issues and questions and ensure swift communication.

Discord on the other hand was primarily just used for making calls as it was something everyone in the group was familiar with. Like Slack, we were able to make channels on discord to have smaller calls and discussions when we divided into teams to tackle specific tasks. Discord being available on all devices was a godsent! The calls on discord were generally (60-90) minutes long and were always fruitful and professional.

Github was our primary system for version control. Being well versed with git, using github is and has always been a pleasure. Along with the version control system, the project board enabled us to organise tasks in a much simpler way. For each spring, we were able to assign users their tasks and then let each task be placed from 'to-do' to 'in progress' to 'done'. The issues section on github allowed us to report bugs and suggest future changes that would help better the project.

Messenger on the other hand was used for general discussions along with other off topic discussions. Things were not kept entirely official in messenger and was used as an area where discussions took place for things not covered during team meetings. Messenger also allowed us to set up and organise the next dates for meetings and team calls.

Last but not least, the most important of all were the face to face meetings done during the semester. In 1920, social psychologist Floyd Allport found that "people worked better in teams even if they weren’t collaborating, competing, or actively communicating with each other". This proved true for us too, working together as a team produced better productivity rates than when people decided to tackle tasks in a solo manner. These, being more personal, facilitated better communication among the team. Being able to be in a room and discuss what we were planning on doing while doing it helped prevent a lot of issues that might have risen if we would have decided only to discuss these later.

While yes, discussions and meetings on topic were very important, a good practice that a well functioning team must have is scheduling time for virtual camaraderie building, including chatting in an informal context. Researchers at MIT’s Human Dynamics Laboratory have found conversations outside of formal meetings are the most important factor that contributes to team success. On various occasions, the team went together for lunch and coffee breaks where the discussions were kept off topic to encourage team building and camaraderie.

**6. Your Project In Detail**

Present a worked example of your *NoseJob* application in action, working upon itself to report on its own code smells. What does your system say about your code? How does this align with your own insights into the code? Is the analysis fair, or useful? What does it get right? What does it get wrong? Feel free to use screenshots here.

If your project requires any non-obvious steps to launch/activate, outline those steps here. We should be able to launch your application successfully from what you tell us here.

**Acknowledgements**

Every team member should contribute an equal effort to this report. Use this optional section to provide report credits, or to highlight a special contribution by a given team member.

**References**

List any bibliographical citations here [optional]