Kaleb Olson / Moussa Diaby
SIES 635
Team Project 2

Pizza Ordering Application

Application for a customer to place a pizza order, and view total, as well as delivery information.

Required Deliverables:

- (5 points) ***Form a team of 2-4 students and select a project***
  Continue with your TP-1 team, or else form a completely new team. In either case, choose to continue implementing TP-1's Skunk, or else select, specify, and implement a different project
- (20 points) ***An application with working Java code***
  Your application should be implemented in Java, use Eclipse for your IDE, and use JUnit for your unit tests. I will consider the possibility of allowing alternative languages, IDEs, and/or other unit testing frameworks to be used instead.
- (5 points) ***Code management using Git and GitHub***
  You must host your entire Eclipse project in a remote repository on GitHub, using regular commits with Git to develop your working code. Your commit messages should be useful and descriptive. All external libraries should be included in your repository, so that I can clone and run your application locally with no additional effort.
- (10 points) ***Comprehensive JUnit tests for all the non-UI code in your application***
  You should provide as complete coverage of your non-UI code via JUnit tests as possible. Your grade for this option will depend on how close you are to 100% coverage.
- (10 points) ***A final team presentation of your project to the class***
  You will give a final presentation to the class on the final day of the course: May 15, 2020. If we're still online by this date, you will instead create a 20 to 30-minute desktop video as part your final presentation and give a 5 to 10-minute overview of your project via Zoom. All team members must participate equally in the presentation and/or video, with a separate handout provided detailed requirements.

Options Selected:

- (5 points) ***Use of some app to provide team communications.*** Some examples are Trello, Slack, and MS Teams, but you can choose others. You'll verify app usage by submitting examples of its use with actual team messages.
- (5 points) ***Use of software tools to provide code metrics***. There are many software tools that can be deployed to measure different attributes or metrics of your project code. Our H5 class activity discusses some of these and a separate document (TP-2.2) will describe others, but you are free to find and use similar tools. You may choose and deploy one or two different tools, measuring your

project code at the beginning, middle, and end of the project. Each deployed tool will count as 5 points towards your optional deliverable requirements.

- (5 points) ***Superior separation of Presentation Logic (UI) and Business Logic (Domain Layer).*** UI code should ideally depend on Domain Layer code, but not the other way around. If you achieve this in your implementation, you will earn up to 5 points depending upon the degree of your separation.
- (5 points) ***Write a use case that describes a fundamental interaction between a user and your SuD.*** It should follow Larman's "fully-dressed" format NextPOS use case, consisting of a main scenario followed by a list of possible extensions (triggers requiring different actions).
- (5 points) ***Draw a domain class diagram (DoCD) that captures the key concepts and their associations.*** Follow the recommendations given in Larman and in class.