The University of Western Ontario
Department of Computer Science
CS2212b 2015

Team Project Acceptance Testing

Your team will attend a one hour meeting with a portion of the marking team. We will test your software and expect your team to represent themselves well in answering questions about the codebase and design decisions.

HELD: March 30th – April 2nd

Functional Testing

We will spend the majority of the hour testing your software.

Setup Process:

1. Clone tagged stage3 submission
2. Build the jar using mvn package
3. Run your program using java –jar

If this process does not work, and cannot be resolved within 5min, the maximum grade you can get on your program's functionality is 45% of the grades allocated to the components submitted for grading.

There will be a time set aside for a dry run of this process on the Friday prior to the submission of stage3.

Testing

We will test your program's adherence to the functional and non-functional requirements, including stated and unstated requirements.

The tests will be thorough, and will include all tests from your user stories as well as tests developed by the marking team. Your team should take the time to test everything you can think of. See Topic 6 for details.

Team Representation

The team will be evaluation on how well you represent your project's code and development. All members should be present unless this is not possible.

We expect professionalism and preparedness.

This meeting intends to mimic interactions you might have with a project manager and/or client. In that regard, interpersonal skill and familiarity with the project, including both its strengths and weaknesses, is essential.

Professionalism

- Take ownership and responsibility for the project
- Highlight positives but acknowledge drawbacks
- Be on time
- Be polite
- Do not interrupt others
- Be actively listening when not speaking
- This is NOT the time to blame people or circumstances
- This is a time to positively represent your software and achievements
- Emotional regulation (i.e. don't get angry, give us attitude)

The following is a list of example discussion points.

- Go over your class diagram, and lead me through how your system works.
- Why did you decide to design your class interactions this way?
- What other GUI designs did you consider, and why did you make the choice you did?
- Can you change the look of __ ?
- Can you change the layout of ____?
- I would have liked to see ____ implemented, how long will adding that component take, and how would you go about it?
- We would like to use a different weather web service ____. Will that be possible and how long will it take you to migrate?