UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

COMPUTER SCIENCE

COS 301 Software Engineering

Capstone Project: Demo 3 Instruction

S. Baror, V. Pieterse, T Mautsa & C. Trivella

1 Introduction

The Capstone Project Demo 3 is **Friday, 19 July 2019**. The Demo's Bookings are available on Google Calendar, use the Google Calendar link provided on the COS 301 portal at the Link section. For Demo 3 we expect a minimum of 60% of the project implemented. You should also pay attention to your Testing policy, Coding Standard and User manual and keep your previously shown requirement documentations in sync with your implementation. Reflect on your collaboration and complete your iPeer before the deadline. There will be no extension after the deadline.

2 Live Demo

Note that the demo slots are 20 minutes. Take into account that you should allow 5 minutes break between teams for wrap up and setup. The actual presentation should be no longer that 15 minutes. **You MUST Demo from your Git Master Branch.** Demos displayed outside your Git Master branch may impact negatively on your marks. During the live demo, you should do the following:

Have a slide show that contains the following.

  - Shows us **bullet list of ALL** your project's requirements.
  - Show us **a list** of what you have done since after Demo 2

- You should show the following live:
  - Implementation of the items on the List you shown in bullet 2 above - at least **Five** use cases different from the use cases you showed at Demo 1 & 2. Ensure that at the end of your Demo, your should have **60% or more of your entire project functionalities completed.**
  - Any other fantastic stuff you have implemented for a bonus of 5 marks?

3 Deliverables

You should maintain all the deliverables of your project. All the appropriate deliverable, as well as the use of the tools, will be evaluated with every demo. The appropriate artefacts should at all times reflect the detail and current state of the project under construction. You should strive to have a working prototype of the implementation available in your master branch at all times.

3.1 COS 301 SE-Git Organisation

Recall that from Demo 2, the COS 301 Git Organisation is your landing page. The Link to the COS 301 SE Git Organisation is also available on the link section of the COS 301 web page. Your team's COS 301 SE Git Organisation should have the following:

1. The Readme should contain a short description of your project.
2. Place **PDF files** of the following in a documentation folder at the top level of your repository and also include links to these documents in your readme.
   (a) A PDF of your functional & Architectural Design documentation in **ONE document**
   (b) A PDF user manual doc
   (c) A PDF of your coding standard and quality doc
   (d) A PDF of Testing policy doc
3. Link to your Project management - scrum board tool of your choice (Waffle, Zenhub, etc.)
4. Use the user profile tools provided by gitHub to setup and maintain your individual profiles of the team members.

3.2 Working prototype

You should implement your system in such a way that you always have a working prototype of the system in your git master branch. The features that are not implemented yet should be mocked. For demo 3 we expect at least seven new use cases to be implemented, tested and dealt with in the user manual. The working prototype must be available on the master branch of your git repo.

3.3 Architectural Requirements and design documentation

This document should be kept in sync with your implemented project.

3.4 Coding standards document

The coding standards document should describe your conventions and styles to ensure a uniform style, clarity, flexibility, reliability and efficiency of your code. Do not duplicate the industry standards you follow. Simply point to them. Focus on describing the process and tools you use to ensure adherence. Also, document the file structure of your repository. See Chapter 18 in the textbook.

3.5 Testing policy document

You should have automated tests. Use a tool such as Travis CI (or any other appropriate tool) to manage and automate testing and deployment of your system. The testing policy document should describe the procedure you are following for testing. Point to your git repository of test cases and test reports. See Chapter 18 in the textbook as well as the documentation of your chosen testing tool.

3.6 User manual

The user manual should start by having a brief description of the project in layman's terms (avoid technical terms). Include a deployment *picture* i.e. something like your UML deployment diagram that should be in your requirements and design document, yet with pretty pictures of the devices you use. Write the document using the guidelines given in the UserManual.pdf document you can find in the Capstone Project folder on clickUP. The detail description of use cases should be only for the use cases that are already implemented (no imaginary or 'we may have' use cases).

4 Requirements and design specifications

This document is a growing document. You should correct errors in the previous version and add more content for every demo.

Pay close attention to improve and correct your SRS document to comply with previously given instructions and feedback you may received.

5 Assessment rubric

Item	Marks
Live Demo	
List all the working use cases you have added since demo 2. Show at least 5 use cases (@ 10 marks each)	
Show other fantastic additions for 5 mark bonus	55
Unit and integration testing (5 marks each)	10
Documentation	
Testing policy	5
Coding standard & quality	5
User manual	10
Corrections to SRS document of demo 2 where mistakes were made	10
Individual	
Quality of reflection in iPeer	5
Total	100

Note that individuals who are identified as social loafers or diligent isolates may be penalized by down-scaling of marks of team deliverables.

6 Project Client

You should have regular discussions/meeting with your project owner (client). Seek their approval of all artefacts that we require for COS 301. Give your client access to your git.

Your client is welcome to attend your demo, but it is not required. Ideally, you should arrange to see your client for an additional half hour or more before or after your demo. The demo itself may be rushed and can not serve as an opportunity for you to ask your client some questions.

If your client needs access to campus – for the demo or any other meeting, please provide the following detail at least **36 hours before the time.**

- Client Name
- Client Email
- Client Phone number
- Date
- Time
- Venue
- Vehicle description
- Vehicle registration number.