UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

COMPUTER SCIENCE

COS 301 Software Engineering

Capstone Project: Demo 4 Instruction

S. Baror, V. Pieterse & C. Trivella

1 Introduction

The Capstone Project Demo 4 is **Friday, 23rd August 2019**. The Demo's Bookings has been assigned, see Figure 1 for the time slots. For Demo 4 we expect a minimum of 80% of the project implemented. You should also pay attention to updating your SRS, Testing policy, Coding Standard and User manual document and keep your previously, clearly labeling the versions. Your documentations should be in sync with your implementation. Reflect on your collaboration and complete your iPeer before the deadline. There will be no extension after the deadline.

COS 301 Capstone Project Demo Schedule - Committee Member Assignment - Friday, 23rd Aug., 2019					
Time slot	**Project name**	**Team name**	**ID**	**Committee Member**	**Project Description**
08:00	Tlou & Botti	None		Linda & Anna	
08:30	Car Classifier	Ctr Alt Elite	17	Linda & Anna	Automatically recognize and classify car make, model and features.
09:00	Amazon Dash	Contrapositive	8	Linda & Anna	A dashboard to show all running instances across all regions, enabling visibility, monitoring and budgetting of resources in an AWS enviroment
09:30	FABI	Nova	13	Linda & Anna	FABI database analytics and interface for data capturing, monitoring and modelling of forest pests and diseases
10:00	A-recognition	Singularity	12	Stefan& Anna	Apply facial recognition and OTPs to allow access based on calendar bookings of boardrooms.
10:30	Botic	Alabama LS	10	Anna & Stefan	An intelligent, privacy aware chatbot.
11:00	Jargon	Syntactic Sugar	9	Anna & Stefan	Real-time Twitter sentiment analysis tool about a user specified topic, feeding data to a live dashboard
11:30	Smart NFC	Vast Expanse	7	Stefan& Anna	Enabling EPI-USE to maximize the use of NFC cards in the office environment for sharing information and managing access and assets
12:00	Realtime fire	Sandwico	3	Stefan& Anna	Save time and money with smart building fire escape routes
12:30	Lunch				
13:30	Ranger app	The Tenacious Technicians	15	Stefan & Hein	A mobile application to assist ERP rangers with various daily tasks such as managing patrol shifts and reporting incidents.
14:00	Wildlife...	DR Bam	11	Hein & Anna	Using live tracking data of wildlife animals to generate a most likely patrol path to ERP Rangers and working out an optimal flight path for accompanying drones.
14:30	Follow me drone	5 Guys 1 Branch	5	Hein & Anna	Identify any dangers in the vicinity of the ranger. It estimates the objects threat level to the ranger and notify them ranger of any detections.
15:00	Coffee Break				
15:30	Taxi boss	Supreme Internect	14	Linda & Hein	Gather data related to the behaviour of taxi drivers for taxi owners to monitor and incent drivers for good driving.
16:00	gyser app	HighTech	6	Hein & Linda	Streamline the process of installing, maintaining and monitoring geysers. Apply smart capturing of data and generate documentation for mainatiance and planning.
16:30	Smart home	MonoToneID	2	Hein & Linda	Manage devices and monitor the engery generation capacity and consumption of a smart home.
17:00	Defendr	Dark nITes	4	Hein & Linda	Using XDP and eBPF to provide (1) ddos protection and (2) load balancing over configurable back-end services.
17:30	Indoor Mall	Brute Force	16	Stefan& Hein	An in-door mall navigation and augmented reality application to assist shoppers in large malls

2 Live Demo

Note that the demo slots are 30 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 than 25 minutes. During the live demo, you should do the following:

Have a slide show that contains the following.

- The core functional requirements of your system (User story-what exactly your system does)
- **List** the most impressive aspect of your project
- Show your deployment diagram picture from your user manual
- Show in **a list** what is left to be implemented and any extension to help your project stand-out on project day.

- You should show the following live:
  - Implementation of the items on the List you shown in bullet above
  - Show the core (most impressive) selection of items of your project
  - Illustrate how you have used automated tests and coding standards to ensure high quality code with minimum bugs.
  - Any other waoooh/fantastic stuff you have implemented since after demo 3
  - Why showcasing your project, mention the knowledge and skills learned in other modules during your degree program that has been applied to your project.

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 off-line after the 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 personal profiles of the team members. **You are required to point to more comprehensive CV's of the members of your team in a separate document in your Git repo or on another profiling platform such as LinkedIn. Your LinkedIn profile should provide your details that employer could reach you) - This will be assessed.**

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	
Working functionalities	60
Testing	10
Implementation and Documentation	
SRS documents	5
User manual	5
Code quality	15
Individual	
Quality of reflection in iPeer	5
Personal profile	5
Total	105

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 **Wednesday before Friday demos. Details of access provided later may not get processed by the security before Friday morning. Email your clients now to RSVP.**

- Client Name
- Client Email
- Client Phone number
- Date

- Time
- Venue
- Vehicle description
- Vehicle registration number.