UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

Department of Computer Science

COS 301
Software Engineering
Capstone Project: Demo 2

Vreda Pieterse and Stacey Omeleze Baror

1 Demo Time

The second Demo is on Friday 13 April. You should book your demo slot for the demo. Bookings will open at 09h00 on Monday 9 April and will close at 09:30 on 11 April.

2 Landing Page

Your landing page is judged on the content and not so much on the style. You should use simple HTML features to display the required content in a consistent and pleasing way. Your landing page should have the following:

- a brief CV for each of the members of the team such as your current position, interests, skills, previous work experiences, and attitudes. You may also link to other profiles such a git profile or LinkedIn profile. Maintain consistency.
- Access to your documentation. Link to a PDF of each of the following documents:
  - Requirements and design document
  - User manual
  - Coding standards document
  - Testing policy document
- Access to your version control and project management tools. Link at least to the following:
  - github repository
  - slack group
  - scrumboard tool of your choice (Waffle, Zenhub, etc.)

3 Demo

You should have a slide show containing at least the following

- A very brief introduction to the team members i.e. name and role in the team.
- A brief description of the project of between 400 and 500 characters (punctuation and spaces included). This description should also appear in the description field which you will find at the top of your landing page editor.
- Git structure (i.e., branching and merging) and code quality.

You should show the following live:

- Implementation of at least five use cases constituting the core functionality of your project.
- Automated tests for the implemented use cases

4 Deliverables

You should maintain the deliverables for your project. All the appropriate deliverables as well as your use of the tools specified in Section 2 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.

4.1 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.

Besides the Implementation code of the working prototype of your project you should also have the following documents:

4.2 Requirements and design documentation

The requirements and design documentation should include the functional requirements, the domain model and the architectural design and structure of your system. See Chapters 4 - 6 in the textbook for reference. You should be able to determine the architectural structural design of your system. For example, is your system based on N-tier, client-server, MVC, Service oriented architecture (SOA), Micro services or another architectural style? Justify your choice.

4.3 Coding standards document

The coding standards document should describe your conventions and styles to ensure an uniform style, clarity, flexibility, reliability and efficiency of your code. Also document the file structure of your repository. See Chapter 18 in the textbook.

4.4 Testing policy document

You should have automated tests. Use a tool such as Travis CI 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.

4.5 User manual

The user manual should start by having a brief description of the project in lay-mens 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 Instructions folder on the CS Web page. 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).

5 Project Client

You should have regular discussions with your client. Seek their approval of all artefacts that we require for COS 301.

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