UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

Department of Computer Science
COS 730
Software Engineering 1

Individual Project: Assignment 2 Instructions

Stacey Baror

1 Introduction

For Assignment 1, you have chosen a system that must contain a minimum of four core components that could be treated as a service. Using the same system, design the software architecture of the system following the rubric structure provided therein, and other materials, especially the textbook **Software Architecture in Practice** *a* book by Len Bass, Paul Clements, and Rick Kazman. The deliverable for Assignment 2 - Software Architectural design is a PDF upload to ClickUp and a VideoDemo to be (uploaded here `https://forms.gle/4RRCMCE7xygvJKtN6`)

An announcement will be made once the demo booking link is available on ClickUP. Ensure to make your demo booking with your Tuks email and add the name of your project as description, booking made without **NOT tuks email** will be removed. Demo slots will be on **(ClickUp - DemoBooking), No Booking No Demo**.

Assignment 2-Software architecture design - a Video Demo presentation of the systems architecture and a PDF upload submission is due on 03 May 2021.

1.1 Administration and Infrastructure

- The deadline for this assignment is **03 May @ 23h30 PM**
- Upload your PDF using ClickUp
- For modelling (UML diagrams), you are required to model the UML diagrams for this assignment and **ALL** future deliverable of COS 730 using any of the tools below.
- Visual-paradigm `https://online.visual-paradigm.com/`
- Or Magic draw `https://www.nomagic.com/products/magicdraw`
- You are always advised to use a proper UML standard syntax to avoid losing marks.
- You should use LATEX to write your documentation. Here is an online LATEX- Overleaf (`https://www.overleaf.com/`)
- At the completion of your SRS document, Upload **ONE PDF** version of your SRS documentation to ClickUp for grading.
- Familiarise yourself with latex, as it is only acceptable form of documentation for COS 730. It is advisable (not compulsory) to integrate your git/github/bitbucket with SRS documentation.
- For grading purposes, ensure you invite me to your git - username: staceybaror, email: stacey.baror@tuks.co.za *(this email is strictly for github - COS 730 queries sent to the email will be ignored)*

2 Software Requirements Specification (architectural specification)

The deliverable for assignment 2 is the SRS document focusing on the architectural design specification for **your chosen system (System xxx)** application, applying the architectural design process.

The document should contain the following sections:

- Introduction
- Non-functional requirements
  - Quality requirements
  - Quantification of the each of the identified quality requirements
- Architectural design
  - Architectural patterns
  - Architectural styles
  - Architectural constraints
  - Actor-system interaction
  - Deployment model
  - Technical requirements (technology)

2.1 Non-functional requirements

The non-functional requirement should specify criteria that can be used to judge the system's operations rather than specific behaviours. It focuses on how to address the **xxx's** application requirements in line with the following subsections:

2.1.1 Quality requirement

Specify each of the quality requirements which are relevant to the system. Examples of quality requirements include performance, reliability, scalability, security, flexibility, maintainability, cost, usability.

Identify the quality requirements of the system, such as availability, security, performance or portability. Furthermore, quantify your choices of quality requirements and state how each of your identified quality requirements will be addressed.

2.1.2 Quantification of the quality requirement

Finally, for each of the quality requirements you have identified in section 2.1.1, quantify the quality requirements relevant to the subsystem. For example, when you identified a scalability requirement - (Scalability is the extent to which a system can handle its growth), you should be able to provide a 'numeric number' of users growth handling number, e.g., the**system xxx payment module** should be able to handle 1,000,000 users per sec.

3 Architectural design

The architecture diagram provides an overview of an entire system, identifying the main components developed for the product and their interfaces.

A system's architecture should focus on the system's conceptual model that defines the structure, behaviour, and views of the system. It should further describe the (architectural patterns) (ii) architectural style and (iii) architectural constraints of the system and justifies why the decisions (i.e., the architectural pattern, style or constraints are appropriate for the system based on your understanding of the architectural structure and domain model. -

3.1 Architectural patterns

An architectural pattern is a general, reusable solution to a commonly occurring problem in software architecture.

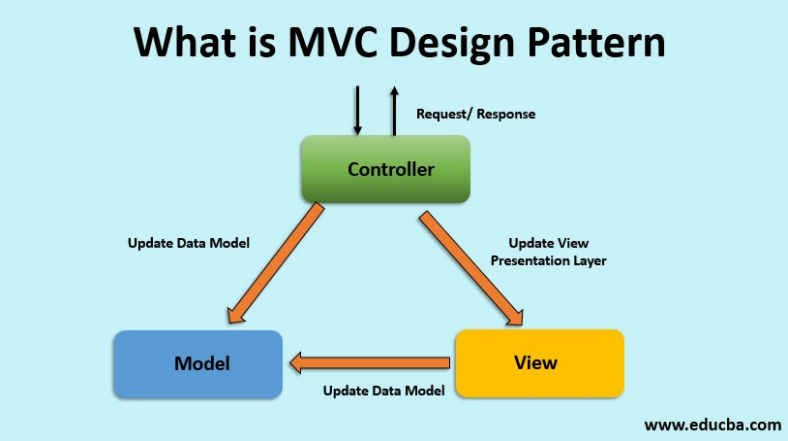


Figure 1: An example of an MVC architecture

Examples (i) Blackboard system (ii) Event-driven architecture (iii) Layers (iv) Micro-services (v) Model–view–controller see example in Figure 1 or (vi) Multi tier architecture (often three-tier or n-tier)

3.2 Architectural Styles (tactics)

"An architectural style is a named collection of architectural design decisions that (1) are applicable in a given development context, (2) constrain architectural design decisions that are specific to a particular system within that context, and (3) elicit beneficial qualities in each resulting system." Every architectural style describes the detailed application of a system's quality requirements quantification, category and structure of the overall system's outlook. **Examples of architectural style; for performance - concurrency, batch processing, or scheduling. Architectural style; for security include access control such as; authentication, authorisation, and audit trail**

3.3 Architectural Constraints

Describe restrictions on design, and alternatives such as restrictions imposed by standards and hardware limitations for the **System xxx** application.

3.4 Actor-System Interaction models

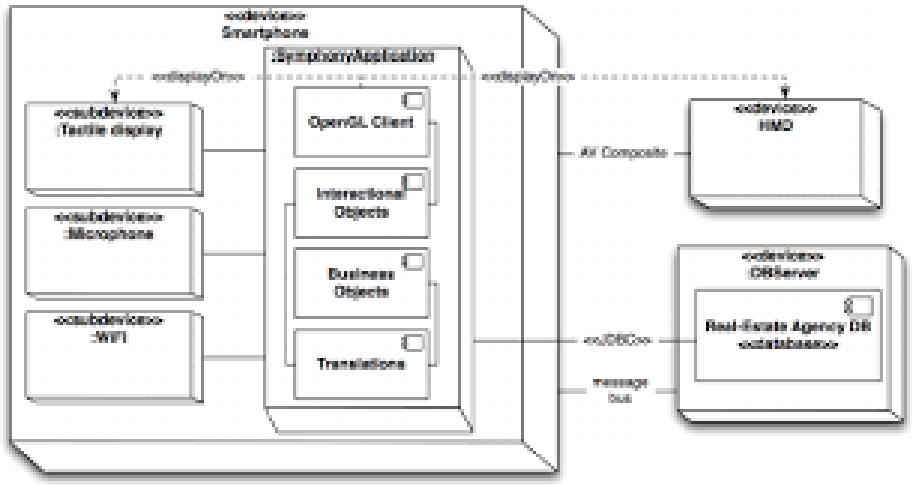
Draw Actor-System Interaction Models for the services that the subsystem should provide for at least one of your choice subsystems. All models should specify preconditions and postconditions. Also, state what exception should be thrown for each precondition if the condition is not met. Figure 2 is an example of an Actor-System Interaction model for an academic management system's main screen.

Precondition: This use case assumes that the staff user has logged into the system and be shown the staff main page.	
Actor: Staff User	System: SAMS
1. TUCBW the staff user clicks the "Add Program" button. 3. The staff user enters program detail and clicks the "submit" button. 5. TUCEW the staff user clicks the "OK" button on the confirmation page.	0. System displays the staff main page. 2. System displays the Add Program page. 4. System checks the submitted info and shows a confirmation message if no error is found.
Postcondition: The added program is immediately available for search.	

Figure 2: An example of an actor-system interaction model

3.5 Deployment Model

A deployment diagram, models the run-time architecture of a system (Figure **??**).



https://en.wikipedia.org/wiki/Deployment$_diagram$

Figure 3: An example of an MVC architecture based Deployment model

It shows the configuration of the hardware elements (nodes) and shows how software elements and artefacts are mapped onto those nodes using technology aligned to the chosen architectural patterns. See Figure 3 for a deployment model for an MVC based system.

3.6 Technology requirements (technical)

Identify various technology and select the best technologies and protocols that you could use in the implementation of the **System xxx** application. To obtain a full mark, you should motivate your technologies choices.

4 Assessment rubric

The deliverable for software architectural assignment 2 is (i) PDF upload (ii) Video upload demo `https://forms.gle/4RRCMCE7xygvJKtN6`

Deliverable	**Marks**
1. Non-Functional (List quality requirements - without explaining is 5mks)	10
2. Architectural design decisions (discuss)	10
2.1 Architectural patterns diagram	20
2.2 Architectural styles	10
2.3 Architectural constraints	10
2.4 Actor-System Interaction	10
2.5 Deployment model (using proper tools, notations & components interaction. Diagram without discussions max mks 5)	20
2.6 Technology requirements (technical)	10
Total	**100**