Object Oriented Software Engineering
Project Work Checklist

Pitching

☐ Project name
☐ Workflow
☐ Objects and interactions

A. Requirements Elicitation

1. Introduction
    ☐ 1.1 Purpose of the system
    ☐ 1.2 Scope of the system
    ☐ 1.3 Objectives and success criteria of the project
    ☐ 1.4 Definitions, acronyms, and abbreviations
    ☐ 1.5 References
    ☐ 1.6 Overview

2. Current system
    ☐ Brief of each workflows (at least 3 workflows)
    ☐ Interaction among objects

3. Proposed system
    ☐3.1 Overview
    ☐3.2 Actors
    ☐3.3 Scenarios
    ☐3.4 Functional requirements
        ☐ Use cases
        ☐ Refined use cases
        ☐ Relationship between actors and use cases
        ☐ Initial analysis objects
    ☐ 3.5 Non-functional requirements
        ☐ Quantifiable measured used
        ☐ Implementation requirements (tools, programming languages, H/W platforms, S/W platforms – high level deployment diagram)
        ☐ Interface requirements (user interface, external API)
        ☐ Operation requirements (administration, configuration)
        ☐ Package requirements (package diagrams)
        ☐ Behavioral requirements and properties (activity diagram at high level of abstraction)

☐ Glossary

B. Requirements Analysis

    ☐ 3.6 System Models
        ☐ 3.6.1 Use case model (use case diagrams, <<includes>>, <<uses>>, <<extends>>)
        ☐ 3.6.2 Object model (identify entity objects, boundary objects, control objects, attributes of objects, draw object diagrams with links and aggregation)
            ☐ 3.6.2.1 Data dictionary of objects

- ☐ 3.6.2.2 Class diagrams with association, generalization, aggregation and composition, visibility of attributes and methods, proper classification methods explained
- ☐ 3.6.2.3 Interaction among objects (collaboration diagram and CRC cards)

☐ 3.6.3 Dynamic models

- ☐ Sequence diagram (for each use case at least 1 sequence diagram)
- ☐ State machine diagram (model behavior of individual objects)

☐ 3.6.4 User interface with navigational paths and screen mock-ups

☐ Completeness, correctness, consistency, reality of the analysis models to be checked.

C. System Design

4. Design of proposed system

☐ 4.1 Identify design goals (in reference to non-functional requirements)

☐ 4.2 Subsystems decomposition with component diagram

- ☐ 4.2.1 Subsystem services
- ☐ 4.2.2 Architecture styles
- ☐ 4.2.3 Mapping subsystems to hardware

☐ 4.3 Design of persistent data management

☐ 4.4 Specification of access control policy

☐ 4.5 Design of global control flow

- ☐ 4.5.1 Activity diagram

D. Object Design

☐ 4.6 Reuse – fit design patterns, frameworks,… into design model

☐ 4.7 Interface/service specification

- ☐ 4.7.1 Signatures, visibility
- ☐ 4.7.2 Contracts – invariants, preconditions, post conditions
- ☐ 4.7.3 List of implementor classes

E. Implementation

☐ 4.8 Design optimization with transformation

- ☐ 4.8.1 Optimizing access paths
- ☐ 4.8.2 Collapsing objects

☐ 4.9 Associations mapping

- ☐ 4.9.1 Collections
- ☐ 4.9.2 Association classes

☐ 4.10 Contracts mapping

- ☐ 4.10.1 Exceptions

☐ 4.11 Storage schema

- ☐ 4.11.1 Mapping classes and attributes
- ☐ 4.11.2 Mapping associations
- ☐ 4.11.3 Mapping inheritance

☐ Glossary