

Project 2 – Blockchain-based Application Design and Development

COMP6452 Software Architecture for Blockchain Applications

1 Introduction

In this project, you will design and develop a blockchain-based application to solve a real-world problem in a chosen domain. You could select a domain from the list of domains proposed in Section 2.1, or select a different domain that is of interest to you. You need to do some research to understand the problems in the chosen domain, identify requirements, design a blockchain-based architecture, and implement a system.

This is a group project with 4 students per group. You are required to give two presentations in week 5 and week 9 as a group. The first presentation should cover the chosen domain and problem, requirement, and high-level architecture. The second presentation should cover the implementation details and a live demo of the proposed system. You need to indicate who did what at the beginning of both presentations clearly. Section 2 gives more detailed instructions.

2 Tasks

2.1 Task 1: Domain Problem and Requirement

1. Select a domain from:
 - (a) Food supply chain
 - (b) Land title management
 - (c) Circular economy
 - (d) Genomics data management
 - (e) Any another domain
2. Discuss why the chosen domain is interesting to you, and the problem is worth solving?
3. Identify at least 3 problems/challenges of the chosen domain that could be addressed using a blockchain
4. Identify at least 3 main functional and non-functional requirements each for the problems/challenges of the chosen domain
5. Post answers to the above 4 questions on the **Project 2 Forum** on Moodle by 10:00 pm on the 3rd Friday of the term. Make sure to indicate the team members too. You will receive feedback from one of the lecturers by the end of week 4.

2.2 Task 2: Architecture Design and Analysis

1. Analyse the suitability of using blockchain technology to address the identified problems/challenges of the chosen domain
2. Give software architecture design to fulfil the identified functional and non-functional requirements
 - Cover design decisions on blockchain platform selection and deployment/roles of blockchain in the system

- Illustrate and describe the software architecture from at least 2 viewpoints + use cases (2 diagrams)
 - Propose a design alternative (1 diagram) and compare the 2 designs against quality attributes
 - Your designs are expected to include interactions with a blockchain with both on and off-chain computational and storage elements
3. Present your findings during the week 5 presentation slot (12-min presentation + 3-min Q&A). All team members must be present. The presentation should cover the following:
- Chosen domain (1 slide)
 - Identified functional and non-functional requirements (1 slide)
 - Analysis of the suitability of using blockchain technology to address the identified problems/challenges of the chosen domain (1 slide)
 - 2 architectural designs and comparison (max 5 slides)
 - Proposed development plan (1 slide)

2.3 Task 3: Development

1. Develop a blockchain-based software system that implements main architecture designed in the Task 2 (Section 2.2). The developed solution should:
- Interact with a blockchain
 - 2 smart contracts with business logic that fulfil 2 functional requirements (not only as data storage)
 - An off-chain computational component
 - An off-chain data storage (e.g., Cloud, Peer-to-Peer data storage, or shared database)
 - An Oracle
 - A GUI is unnecessary as long as the communication between components is clearly demonstrated
2. Present your solution during the week 9 presentation slot (15 minutes). All team members must be present. The presentation should cover the following:
- Solution design and implementation (2-3 slides)
 - Demo

3 Project Submission

Deliverable	30 points in total
Presentation 1 (Domain Problem + Requirements + Architecture Design)	4 + 4 + 8
Presentation 2 (Implementation details + Source code + Live Demo)	8 + 6

The slides, source code including unit tests, and address of smart contracts need to be submitted to the course Moodle page after the presentation sessions. Include a `readme.txt` file with specific instructions and libraries that may be needed to run your program. Upload the source code, `smart_contract_addresses.txt`, and `readme.txt` as a single `.zip` or `.tar.gz` file.

Plagiarism checker will be used to analyze the submitted code and answer for open question (changing the name of state variables will not help). The UNSW has an ongoing commitment to fostering a culture of learning informed by academic integrity. All the UNSW staff and students have a responsibility to adhere to this principle of academic integrity. Plagiarism undermines academic integrity and is not tolerated at the UNSW.