

Applied Development Operations (DevOps)

Coursework Semester 2: 20/21

**A Digital version of the Coursework should be submitted via Vision.
no later than 11:59pm on submission dates**

Important: Read through this document carefully before you begin to work on your assignment.

Course Lecturer Contact Details

Abrar Ullah (Dubai) Email: a.ullah@hw.ac.uk

Overview

This coursework contributes 70% of your mark for the course (the other 30% comes from two class tests). The coursework involves design, development, staging and deployment operations of an application. The application has been chosen to be complex enough to enable you to demonstrate understanding of software development architectures, security design and automation, continuous integration (CI), continuous development/delivery (CD), continuous monitoring and operations in pre-production and production environments.

The scenario is based around a patient information system in a busy tertiary care hospital.

You should work in groups of 4-5, with the group collaborating over the design and implementation of the application. Larger groups will be expected to achieve proportionally more. All group members are required to contribute to design, implementation and report writing. To ensure fairness, each group member will be asked to assess the contribution of other members using a peer review form. If there are significant differences between levels of contribution, then marks will be adjusted accordingly.

Project Outline

You are assigned a project to develop a system for a tertiary care hospital that serves approximately 2000 patients daily. There are three entry routes for patients depending on their health conditions. Patients needing urgent attention are admitted to the hospital through Accident and Emergency (A&E) Department. Patients visit the Outdoor Patient Department (OPD) for their routine checkup on a pre-booked appointment or on a first come first serve basis.

There is a total of 36 wards which are operational under 9 departments including Medicine, Surgery, Orthopedics, Pediatrics, ENT, Ophthalmology, Gynecology, Dermatology and Oncology. Each department has 4 wards e.g., Surgery Department has Surgical A, Surgical B, Surgical C, and Surgical D. There are two special wards for critical patients i.e., Cardiology Care Unit (CCU) and Intensive Care Units (ICU). Patients that require monitoring, care and specialized treatment are admitted to the above wards through OPD or A&E.

Outdoor patients visiting the hospital through OPD and A&E or indoor patients admitted to wards can be referred to use different diagnostic and treatment services available at the hospital premises including Radiology, Pathology, Physiotherapy, Blood Bank, and Operation Theatres.

The system has a service integration interface to the health regulator and Government for patient data audit and analysis. The system also provides a service interface to affiliated medical Universities for the undergraduate and postgraduate medical research students. The data provided to research students is anonymized with restricted access control and audit.

The system should cover the following key functions:

- 1) User Authentication
- 2) Initial Patient Registration
- 3) Ward Admissions
- 4) Diagnostic Lab (Pathology) Registration
- 5) Laboratory Results Integration
- 6) Patient Treatment
- 7) Patient Referrals
- 8) Radiology Lab Registration
- 9) Radiology Lab Results Interface

At different entry points, patients are registered by the operational staff. The treatment charts, daily temperature, patient intake/output, progress report, referrals are managed by medics including doctors, nurses, paramedics, and technical staff.

You are part of a group of 4-5 students hired from the Computer Science departments of the universities and assigned to:

- Design the system architecture and propose a viable software development methodology.
- Develop, implement, and deploy **at least three modules** of the system.
- You are expected to select an appropriate software develop methodology, software architecture, change management and automated deployment & operations. The project is undertaken in two phases including Dev Phase and Ops Phase.

What do you need to do

The application is developed and deployed in **two phases** and your group is expected to complete the following phases:

- **Dev Phase** - follows on from materials you will be taught in Part 1 of the course, and concentrates on planning, software architecture, software engineering methodology, change management, code management, security, and testing process.
- **Ops Phase** - follows on from materials you will be taught in Part 2 of the course, and focuses on staging, deployment, implementation of containers, virtual machines, and deployment to standard, virtual and cloud environments.

The following table shows when each of these deliverables is due, and how much they are worth:

Phase	Deliverable	Weight	Deadline
Dev - Phase	Group Report 1 and Dev Code	35%	Sunday 28 th Feb (Week 7)
Ops - Phase	Group Report 2 and CI/CD implementation and Presentation	35%	Wednesday 31 st Mar (Week 12)

Dev Phase

The Dev Phase is worth 35% of Module mark available

Dev phases is designed to assess your understanding and knowledge of software development methodology, change management, software architecture, security design and testing, code management and version control.

You are required to design a simulation of selected modules of a patient information system for a tertiary care hospital. This first stage develops the design, architecture, team and code management process, and code control of a total of three modules from authentication (mandatory), OPD and A&E registration, hospital admissions, treatment, diagnosis registration, operation theatre, and discharge and referral.

Dev Phase Deliverables

You must provide an appropriate project plan, software engineering methodology, software architecture with justification, code management plan, change management process and implementation, and a comprehensive plan for testing of the Patient Information System. You should complete the report as you go through the different stages e.g. design, architecture, methodology, change process etc. Your report should be succinct (no more than 5000 words) and you should use appendices for any additional information.

You must also provide a brief description and steps to explain - how to open and run the code.

You must submit your code through vision with a link to Git repository. The system must meet the following requirements.

Requirements (Dev Phase)

For Dev phase, your application will provide a complete design architecture, development approach, team organization, **authentication** and **two additional modules** of a patient information system. In addition, you will write a report specified in the report section. The hospital system modules are described below.

1. All staff including registration clerks, doctors, nurses, paramedics must login to the system to perform different actions specified below. These roles have specified access rights to different sections.
2. Only the registration clerk can register a patient into the system at different service points in the hospital. At this point only basic personal details and known diseases or complains are entered in the system.
3. Doctors, nurses, and paramedics can retrieve patient's details using a unique patient identifier and populate additional diseases and referral details at OPD, A&E and other service points.
4. Doctor can refer patients to various services including Radiology, Pathology, Blood Bank, Physiotherapy, Operation Theatre, ICU, CCU and Wards.
5. Patients can be registered at any of the service points above in serial 4. by the registration clerk.

6. The Pathology and Radiology results for patients are automatically imported in the system via an API link from various diagnostic machines e.g., digital X-ray, CT-scan, MRI, PCR etc.
7. For indoor patients admitted to a hospital ward or facility, an on-duty nurse enters record of patients, hourly temperature, blood pressure, and pulse rate. This record is transferred to the system automatically via connected devices.
8. Nurse also records patients medicine and treatment details including all intake and output information with date and time.
9. Doctor records diagnosis, patient treatment, prescriptions, and daily progress manually or via audio which is automatically transcribed.
10. A doctor signs off discharge summary and home treatment plan for an outdoor or indoor patient.

Other Requirements

These are the software engineering requirements for Dev Phase:

1. Your application should be implemented using NodeJS and MongoDB.
2. There is no requirement for developing a user interface for the modules. However, your system should accept user input through various API endpoints.
3. The system should produce output using JSON format through various API endpoints for basic information and reports.
4. The authentication and access control should be implemented through an API interface.
5. Using version control is a mandatory requirement. Your group should set up a repository, and a link to this repository should be included in your report.
6. Integrate the version control software with JIRA or a Project and Change Management Software of your choice. The process from change management to code management should be automated and described in the written report.
7. Use an agile methodology and provide details for how change management and continuous development is implemented.

Group Report

The report will consist of several sections:

1. Names of group members
2. A link to your repository.
3. An explanation of the change management strategy and implementation.
4. A suitable system architecture, visual representation of the design, and targeted deployment environment with critical analysis and justification.
5. Steps explaining how to deploy your project developed in NodeJS and MongoDB.
6. Explain, how your chosen tools, software development methodology and version control support change management, continuous development, and continuous testing.
7. Provide a list of all functions and endpoints in your written report.

8. Provide implementation details of agile methodology.

Try to keep the writing succinct. Your report should be **no more than 5000 words**.

Your group report and application for Dev Phase will be marked according to the following criteria:

Criteria	Wt	A (70-100%)	B(60-69%)	C(50-59%)	D(40-49%)	E/F (<40%)
Development and Architecture Design	30%	Clear well thought out, well justified architecture design. Excellent visual representation. Demonstrate excellent understanding of different architecture types	Mostly clear and justified architecture design. Good visual representation and description	Some issues, with the architecture design and visual representation. Some flaws in design description.	Significant issues with the architecture design.	No real indication or understanding of system architecture design
Functionality	30%	All requirements have been met to a high standard	Most of the requirements are met	A number of requirements are incomplete	Significant limitations in functionality	Key requirements are not met. Very little achieved
Change Management Implementation	25%	Effective use of change management concepts and application	Most of the change management concepts and application are implemented	Some issues with implementation of change management concepts and application	Significant limitations with implementation of change management concepts and application	Very little evidence of change management implementation
Authentication and Access Control	5%	Effective use and implementation of access control and authentication	Most of the authentication and access control requirements are met	Some issues with implementation of authentication and access control requirements	Significant limitations with implementation of authentication and access control requirements	Key authentication and access control requirements are not met. Very little achieved
Version Control	10%	Effective use of version control integrated with Agile application	Most of the version control concepts and Agile application implemented	Some issues with implementation of version control concepts and Agile application	Significant limitations with implementation of version control concepts and Agile application	Very little implementation of version control and little evidence of Agile implementation

Submission

All reports and code should be submitted through Vision using the links provided in the Assessment section of the course. Late submissions will be marked according to the university's late submissions policy, i.e. a 30% deduction if submitted within 5 working days of the deadline and no mark after that. If you have mitigating circumstances, please submit the form available at: <https://www.hw.ac.uk/students/studies/examinations/mitigating-circumstances.htm>

Note that all submissions will be checked for plagiarism.

Ops Phase

Ops phase is designed to assess your understanding and practical knowledge of virtualization, containerization, continuous integration, continuous testing, continuous delivery, and continuous monitoring. In this phase, your group will use the application developed at the Dev Phase to simulate implementation of DevOps concepts using appropriate tools for development, change, testing and deployment using containers and CI/CD tools.

Operations Requirements

These are the functional requirements. These requirements cover simulation of design and implementation of DevOps operations.

1. The simulation consists of a software development environment setup using version control and pipelines implementation.
2. Using application from Dev Phase, configure Dockers container for your development environment and code.
3. Design and implement the Continuous Development and Continuous Testing.
4. Design and implement Continuous Integration, Continuous Delivery Pipelines.
5. The pipeline should integrate with JIRA or a tool of your choice for Agile implementation.
6. The pipeline should implement the Security testing strategy.
7. The pipeline automation should automate a full cycle of DevOps operations.

Other Requirements

These are the recommended software tools for implementation of DevOps concepts:

1. Your operations should be implemented using Git, Dockers, Jenkins, and Kubernetes.
2. Provide relevant export files, links, and credentials as part of your submission.

Group Report

The group report for Ops Phase should include the following:

1. An explanation of DevOps strategy with critical analysis and implementation plan.
2. Provide critical analysis and recommendation for using containerization, virtualization, and cloud environments for various systems
3. An explanation of CI/CD plan with details about the core pipelines and a step-by-step procedure for DevOps implementation.
4. An explanation of how each step of the pipeline is implemented with justification and technical guidelines
5. An explanation about continuous testing, security strategy and security implementation in the CI/CD pipelines
6. An explanation and critical analysis of continuous monitoring in DevOps

Try to keep the writing concise. Your report should be **no longer than 25 pages**.

Group Presentation

The Group will be asked to present their DevOps process and implementation in 15 minutes scheduled time.

Your group report and implementation of Ops Phase will be marked according to the following assessment criteria:

Criteria	Wt	A (70-100%)	B(60-69%)	C(50-59%)	D(40-49%)	E/F (<40%)
DevOps strategy	10%	Clear well thought out plan and well justified DevOps Strategy with implementation steps and visual representation	Mostly clear and justified DevOps Strategy and plan with implementation steps	Some issues, with the DevOps strategy and implementation on strategy.	Significant issues with the DevOps strategy and implementation on strategy	Impractical or no real strategy and unrealistic implementation plan
Containerization and Virtualization	20%	Clear understanding of virtualization and containerization in DevOps. Effective implementation of containers in Dockers.	Good understanding of virtualization and containerization in DevOps. Implementation of containers in Dockers.	Some issues with Docker implementation and analysis of containers and virtualization in the context of DevOps.	Significant limitations Docker implementation and analysis of containers and virtualization in the context of DevOps.	Docker implementation requirement not met. Very little or no analysis of containers and virtualization in DevOps.
Continuous Integration & Continuous Delivery Implementation	35%	Effective use of CI/CD concepts. Flawless implementation of CI/CD pipelines with step-by-step technical guidelines	Most of the CI/CD concepts and application are implemented. Technical guidelines included	Some issues with implementation of CI/CD concepts and application. Issues with Technical guidelines	Significant limitations with implementation of CI/CD concepts and application	Very little evidence of CI/CD implementation
Continuous Testing and Security Implementation	25%	Clear and well thought testing and security strategy. Effective implementation of continuous testing	Generally, a good testing and security strategy and implementation of continuous testing in CI/CD pipelines	Some issues with testing and security strategy and implementation of continuous testing in CI/CD pipelines	Significant limitations with testing and security strategy and implementation of continuous testing in CI/CD pipelines	Poor testing and security strategy with inadequate or no implementation in CI/CD pipelines
Continuous Monitoring	10%	Clear and well thought continuous monitoring plan. Excellent critical analysis of monitoring systems and implementation	Generally, a good continuous monitoring plan. Good understanding of monitoring systems and implementation	Some issues with continuous monitoring plan. The plan lacks some clarity and details	Significant limitations with continuous monitoring plan.	Little or no plan for continuous monitoring. Poor understanding of monitoring concepts.

		strategy justification	with	on strategy with justification			
--	--	---------------------------	------	--------------------------------------	--	--	--

Submission

See instructions for Dev Phase submission.