NITTE
(Deemed to be University)

NMAM INSTITUTE
OF TECHNOLOGY

Course Name:Information Science and Engineering

Title:Campus Placement Portal

Student Name:Akshatha Nayak

USN:NNM24IS018

Semester&Section: 4’A’

Branch: Information Science and Engineering

Institution Name: NMAMIT Institute of Technology

Academic Year:2026

Faculty Name: Dr. Jason Elroy Martis

Date of Submission:12-02-2026

Abstract:

Campus Placement Portals are a bridge between students, placement organizers, and recruiters by digitizing the entire hiring workflow in colleges. It enables students to create profiles, upload resumes, view eligibility, and apply to drives, while organizers manage job hiring, shortlisting, schedules, and communication. Recruiters use the platform to publish requirements, shortlist candidates, and conduct selection rounds perfectly. The case study examines an existing campus placement portal from a software engineering point of view by highlighting its key aspects, stakeholders, data flow, and operational challenges such as data accuracy, scalability during peaks, safety of student information, and impartial eligibility enforcement. The report also shows comparison between Waterfall, Incremental Development, and Spiral process models to examine a suitable approach for developing and enhancing such portals, followed by a functional and non-functional requirements document and a validation strategy.

Github Repository Link:

https://github.com/akshathan80-max/NNM24IS018_task1

Introduction:

Campus placements are indeed a sensitive process in colleges, involving multiple stakeholders such as students, placement incharges, working staff, and recruiting companies.The system must remain dependable during peak periods such as drive registrations and shortlisting deadlines.
In the case study. The report also shows the comparison between the three models, Waterfall, Incremental Development, and Spiral models to help us understand which model suits development and continuous improvement of such a portal.

Problem Statement:

As placement activities keep on increasing, institutions are facing challenges in managing accurate student data, enabling eligibility criteria accurately, handling heavy traffic during drive registrations, and maintaining privacy of sensitive information. Recruiters expect timely shortlists and proper communication, while students need transparent status updates and a smooth application experience.

The problem addressed in the case study is to understand how an existing campus placement portal is able to handle these operational and technical challenges and identify suitable development practices to improve scalability, security, and usability.

System Description:

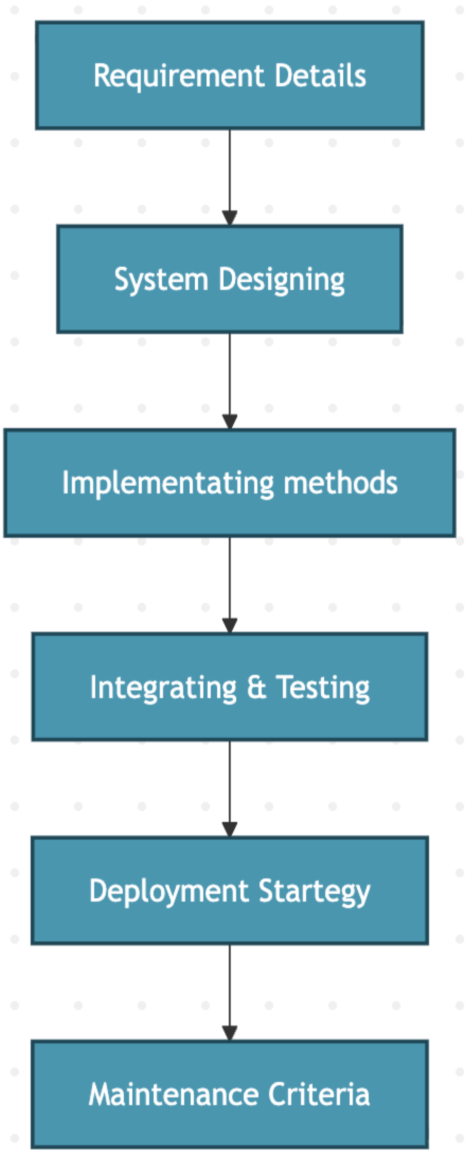
A campus placement portal usually follows a client–server relationship. The client side includes web/mobile interfaces used by students, placement coordinators, and recruiting incharge. The server side provides APIs and business logic for managing profiles, checking eligibility , analyzing job postings, and reporting.Coordinators schedule selection rounds (test, interview, HR Meeting) and send mails. The portal tracks the status of every student , stores results, and produces reports for departments and placement offices.

Tools and Technologies used:

Documentation tool: Microsoft Word
Diagrams: Draw.io
Repository hosting: GitHub
Similarity check: Drillbit

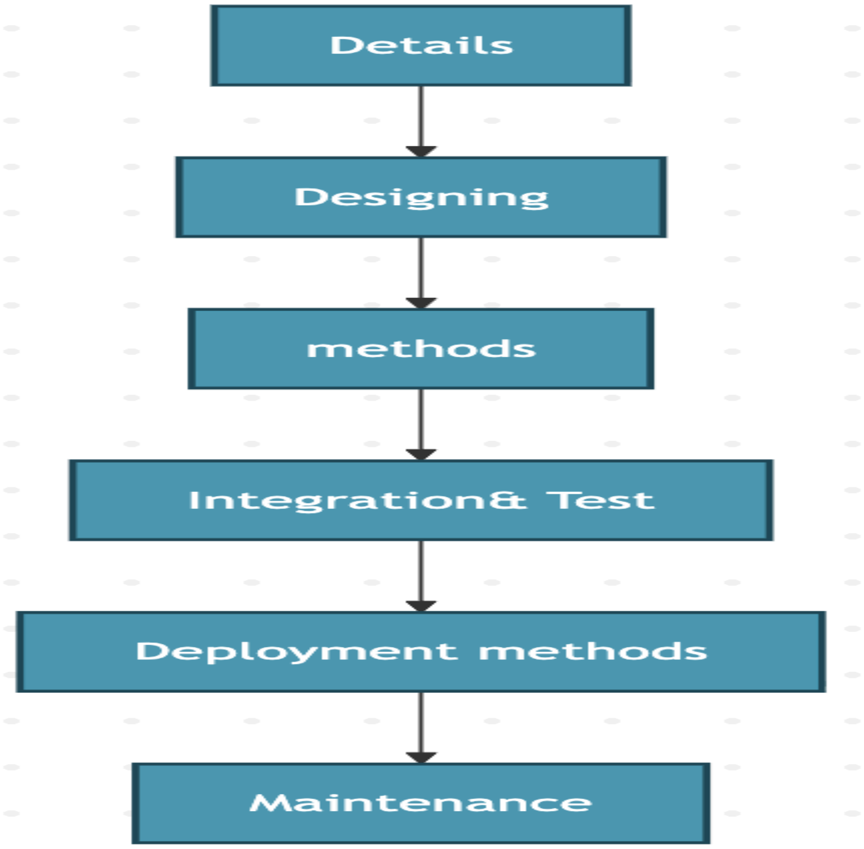
Analysis and Comparison:

The Waterfall Model is a linear sequence where each phase is completed before the next begins: requirement, design, implementation, testing, deployment, and maintenance. For the campus placement portal, this approach becomes challenging because requirements keep changing—new company formats, updated eligibility rules, resume templates, frequent rounds, and policy updates. In Waterfall, late changes can be costly because they may require redesign and re-testing of large parts of the system. Risks include peak-load failures during registrations or privacy and security issues may be discovered lately, which can affect trust and disrupt smooth operations. Therefore, Waterfall Model is generally considered less suitable for placement portals that require frequent improvements and policy-driven updates



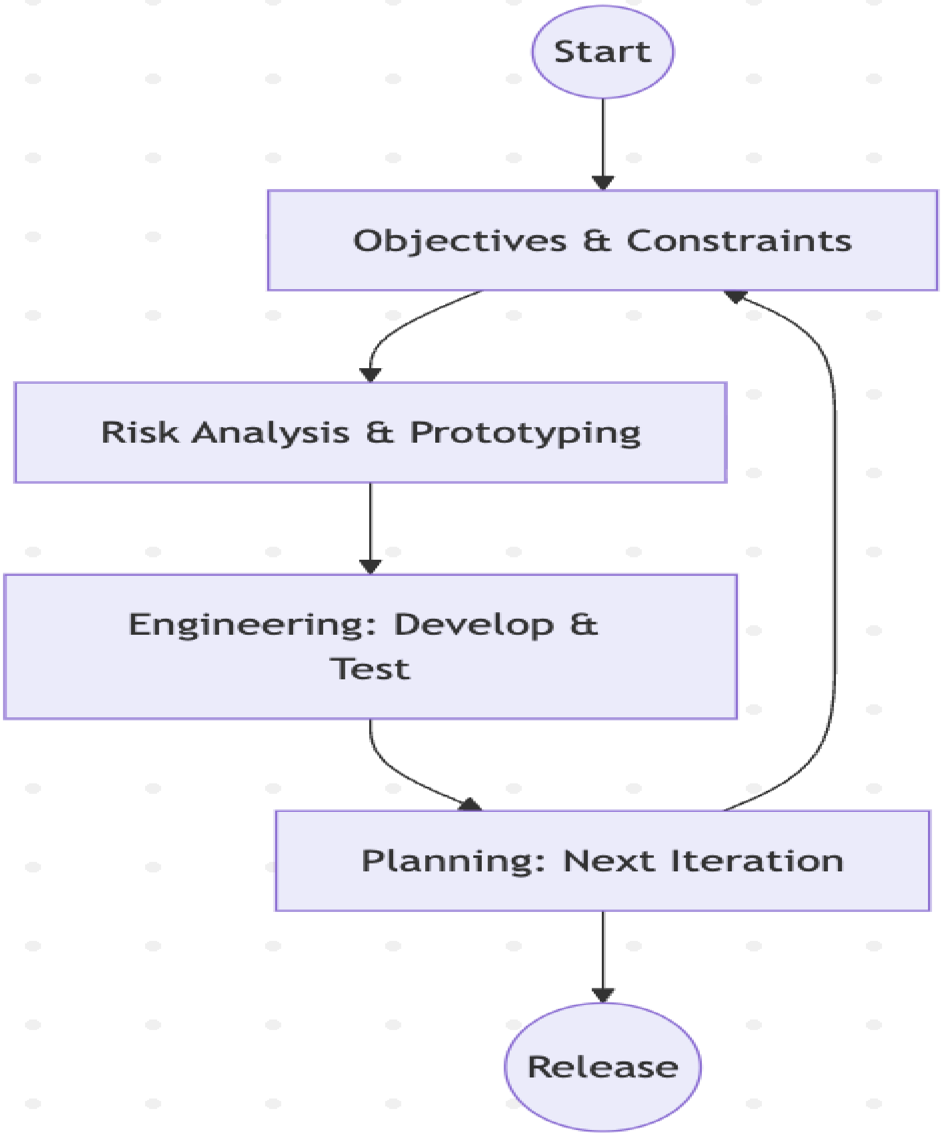
Incremental Development Model :

The Incremental Development Model builds the system in a small, reliable design. We test each increment and review it, allowing coordinators and students to provide feedback easily. This model supports continuous enhancing without disturbing the entire system, reduces errors, and is effective for universities. Because placement portals help in evolving every semester, incremental development is typically a practical choice.



Spiral Model Analysis:

The Spiral Model combines iterative development with explicit risk analysis in every cycle. For a placement portal, key risks include leakage of student data, incorrect eligibility filtering, failures during high traffic, and integration issues with email/SMS services. Spiral is useful when the system must handle sensitive data and high reliability requirements, because each loop focuses on identifying risks, prototyping solutions, validating them, and then building the next version. However, Spiral can be more time-consuming and expensive because it needs careful planning and expertise. It is best suited if the portal is large-scale or must meet strict security and compliance requirements.



Requirement Document:

Functional Requirements:

Students should be able to register and login securely and manage their profile.
Students should be able to upload and update resumes and required documents.
Recruiters and coordinators must be able to create job postings with eligibility criteria.
The system should have to automatically check eligibility and permit only allowable applications.
Students should be allowed to apply for drives and view application status.

Recruiters and coordinators must be able to filter applicants and generate shortlists.
The portal should support scheduling of rounds and sending notifications to candidates.
The system should maintain audit logs for critical activities (shortlisting, status updates).

Non Functional Requirements:

The portal should be able to handle peak loads during deadlines with acceptable response time.

Students' personal data and documents must be stored and transmitted safely.

The system should have availability during placement season.

The portal must be scalable to support an increasing number of students and placement drives.

The UI should be efficient to use on both mobile as well as desktop screens.

The system should maintain data integrity (no fake or wrong information).

Requirements Validation Strategy :

Requirements can be validated by reviewing them with stakeholders: students, coordinators, staff, and recruiters. A sample of key screens (profile, job listing, apply, shortlist, schedule) should be enacted and feedback should be recorded. Each addition should include test cases opposite to requirements, such as eligibility testing, application submissions, and notification transportation. Pilot runs with a small group of students and one recruiter can validate real workings before full deployment conditions.

Challenges in Requirements Engineering:

Eligibility rules and placement policies keep changing frequently.
Different recruiters are in need of different job post formats and selection workings.

Ensuring accuracy and impartiality in shortlisting logic is a challenging topic.
Protecting sensitive student data while providing access to authorized recruiters.
Handling heavy traffic during registrations and shortlist announcements.
Handling usability improvements with time and budget problems.

Result and Analysis :

Based on this case study, Incremental Development is indeed suitable for a campus placement portal because the system evolves continuously and rapidly with varying recruiter requirements and college rules. Delivering the portal in addition ensures early deployment of core features and gradual addition of advanced modules such as testing, automating , and improving security, while incorporating stakeholder feedback.

Conclusion :

A campus placement portal highlights the interaction between students, recruiters, and placement coordinators by managing their profiles, job postings, checking their eligibility , applications, shortlisting, scheduling, and communication. The system must always remain reliable and secure because it handles private student information and experiences heavy usage during placement drives. This case study compared Waterfall, Incremental, and Spiral models for developing this kind of portal. Incremental Development is a best choice for most institutions due to frequent changes and the need for frequent improvement, while Spiral can be applied for dangerous components such as security, privacy, and peak-load handling

References:

Roger S. Pressman and Bruce R. Maxim, *Software Engineering: A Practitioner’s Approach*, McGraw‑Hill Education, 8th ed., 2014 . (accessed Feb. 12, 2026).

Software Development Life Cycle (SDLC),*IBM Documentation*, https://www.ibm.com/topics/sdlc (Accessed: 12‑Feb‑2026).