DrillBit

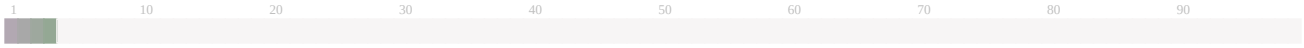
The Report is Generated by DrillBit Plagiarism Detection Software

Submission Information

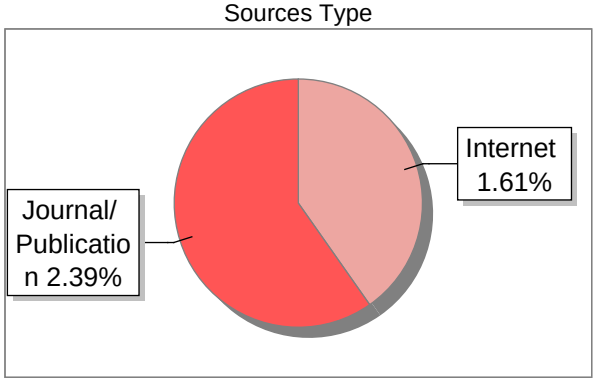
Author Name	Anvitha Rao
Title	Task
Paper/Submission ID	3335607
Submitted by	nnm23is030@nmamit.in
Submission Date	2025-02-16 14:51:51
Total Pages, Total Words	18, 2726
Document type	Project Work

Result Information

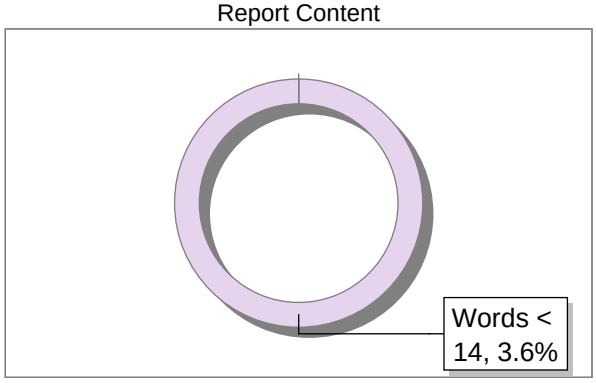
Similarity **4 %**



Sources Type



Report Content



Exclude Information

Quotes	Not Excluded
References/Bibliography	Not Excluded
Source: Excluded < 14 Words	Not Excluded
Excluded Source	**0 %**
Excluded Phrases	Not Excluded

Database Selection

Language	English
Student Papers	Yes
Journals & publishers	Yes
Internet or Web	Yes
Institution Repository	Yes

A Unique QR Code use to View/Download/Share Pdf File

DrillBit

DrillBit Similarity Report

4	9	A
SIMILARITY %	MATCHED SOURCES	GRADE

A-Satisfactory (0-10%)
B-Upgrade (11-40%)
C-Poor (41-60%)
D-Unacceptable (61-100%)

LOCATION	MATCHED DOMAIN	%	SOURCE TYPE
1	degruyter.com	1	Internet Data
2	www.scribd.com	1	Internet Data
3	175.107.63.117	<1	Publication
4	www.aashtoware.org	<1	Publication
5	en.wikipedia.org	<1	Internet Data
6	touroscholar.touro.edu	<1	Publication
7	actascientific.com	<1	Publication
8	Temporal impact analysis and adaptation for service-based systems by Saralaya-2019	<1	Publication
9	IEEE 2018 7th International Conference on Reliability, Infocom Technologies an by Wadhahi-2018	<1	Publication

NITTE (Deemed to be University) | NMAM INSTITUTE OF TECHNOLOGY

INTRODUCTION TO SOFTWARE ENGINEERING

IS1103-1

TASK-1

Analysis of the software development lifecycle (SDLC) of a real-world system by conducting a comparative study of process models and their impact on requirements management.

Topic: IRCTC Train Booking System

Submitted by: Anvitha V Rao **USN:** NNM23IS020

Submitting to:

Dr. Jason Elroy Martis
Associate Professor
Department of Information Science and Engineering
NMAM Institute of Technology, Nitte

Publishing:

This paper is hosted on a GitHub repository-

-

INDEX

1.Introduction

The IRTC, Indian railway catering and Tourism Corporation is the system of ticketing for the Indian Railways. The Indian railway catering and Tourism Corporation serves lakhs of passengers every day. It also offers the reservation of tickets, cancelling of the tickets if not needed for them and also gives the information about the seat availability checking. Also, it gives the information about the update status of PNR status. In addition to this it also tells us about the tourism and catering packages. The system must be highly scalable, secure and efficient due to the vast user base and also the high transaction volumes in the application. It must meet the above requirements in order to meet the dynamic needs of Indian railway passenger.

The IRCTC train booking system allows the users to book the train tickets from the online through the web-based interface and mobile applications. This application helps in the integration of multiple modules, which includes user authentication, information regarding the train schedule, the backend reservation system and the gateway payments which ensures the seamless ticketing operations.

2.Analysis and Comparison

The SDLC, Software development life cycle is the structured procedure for the development of the software application. This SDLC is used for defining the different phases that are involved in software development, which ensures the efficiency, quality and also reliability. The SDLC models are listed below:

2.1 Waterfall model

2.2 Incremental model

2.3 Spiral model

Here, there will be a detailed comparison of the Software development life cycle models, that is waterfall model, incremental model and spiral model also there will be highlight on their suitability based on:

a) Functional and non-functional requirements.

b)Risk and change management.

c) Time and cost constraints.

2.1 Waterfall model:

Waterfall model is a sequential and linear approach in which each stage or phase must be finished fully before moving to the next phase of the system. It includes the following

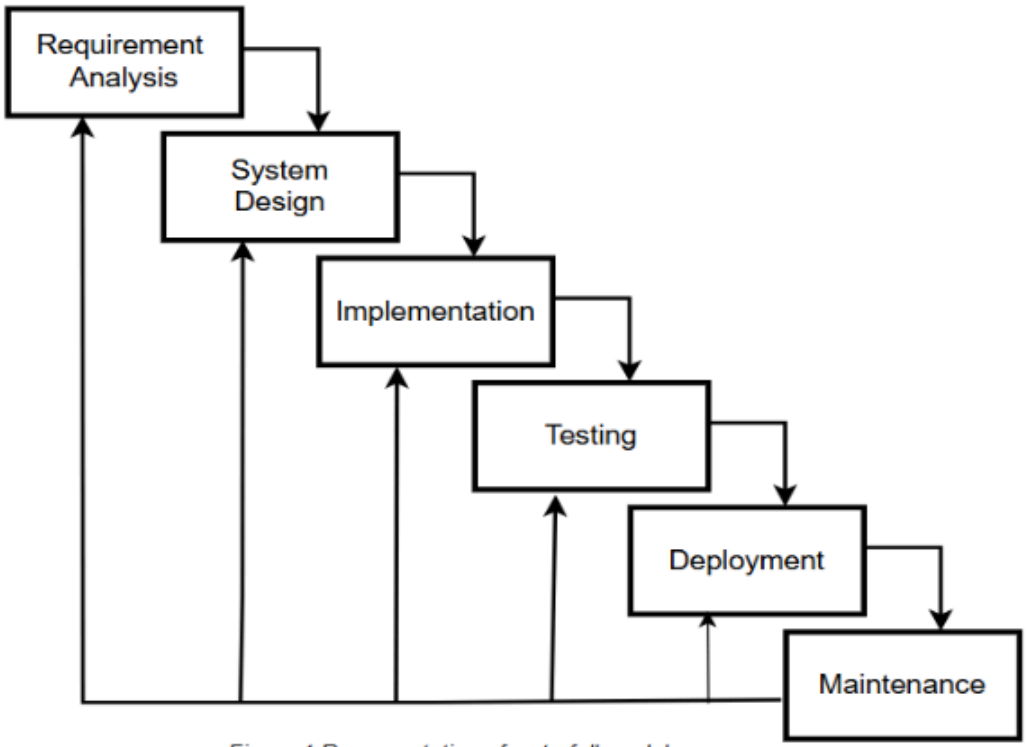


Figure 1:Representation of waterfall model

Phase-1: Requirement analysis

In this first phase of requirement analysis, all non-functional and functional requirements of the systems will be gathered, analyzed and documented in the SRS, Software Requirement Specification document from IRCTC system. Several people and Stakeholders, which also includes clients and developers will be collaborated together to ensure the completeness in this project. Here there will be nodding or design, the main focus here is on understanding the needs of the IRCTC system and constraints of the system and the proper functioning of software.

Phase-2: System design

System design and architecture are created in this phase on the basis of

Software requirement specification. Some of the tools like ER models, UML diagrams, data flow diagram s are often used to design the software system.

Phase-3: Implementation

In the third phase the actual software development of the IRCTC ticket booking system takes place takes place. The programmers will translate the system design into the source code by using the proper and appropriate programming language that ensure the smooth functioning of the IRCTC train ticket booking system.

Phase-4: Testing

In the fourth phase of waterfall model, the main task is to ensure the software of the IRCTC system that is developed functions correctly and meets the specified requirement of the client and the users functions smoothly. This includes integration testing, acceptance testing etc.

Phase-5: Deployment

In the fifth phase, once the testing phase is finished successfully, the completed software is deployed to the real world. Real world means to the railway head and the users. The feedback from the people using it will be later collected in the next phase.

Phase-6: Maintenance

In this phase the deployed software is kept checking by the developer if there are any issues like bugs etc. present after the usage by the client's environment. Like if the IRCTC application or software does not function correctly, the software development team checks for the fault and improves the system.

2.2 Incremental Model:

Incremental model is a type of SDCL model in which the process of software development processes into small and manageable modules which are called as increments. Here each increments adds the fresh functionality and improves the previously existed versions.

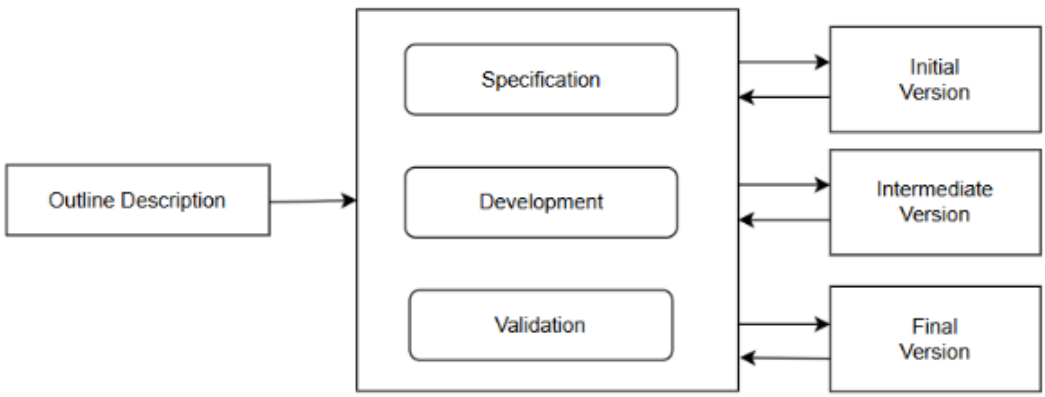


Figure-2:Representation of Incremental model

Phase-1: Requirement analysis

All the core and additional requirements of the IRCTC Ticket Booking System are collected in this phase. For the first increment, the core features are prioritized. The phase is mainly concentrated on adding the additional requirements for the system of the train booking.

Phase-2: Design

Here the system architecture is designed incrementally which ensures that each module integrates perfectly with the previously existed system. Also, it improves the system design that previously existed in a better and good way.

Phase-3: Implementation

User registration and train search functionalities comes under the implementation which is the first increment. The backend works by

using Python, Java and frontend works by using React or Angular for the good and smooth functioning.

Phase-4: Testing

Before deploying, each increment of the software undergoes perfect testing. Types of key testing includes the testing of: unit, integration, System, Security.

Phase-5: Deployment

After successful testing, the completed software is deployed to the real world. This allows users to search and register for the train. Each newly created increment undergoes through the process of testing and followed by deployment to ensure the integration.

Phase-6: Maintenance

Maintenance phase is attained after all the increment phases are deployed which ensures continuous updated and smooth operation. Maintenance includes: Corrective maintenance, Adaptive maintenance, Perfective maintenance.

2.3 Spiral Model:

In the model of Spiral, risk assessment and iterative development through multiple cycles are done in the spiral model.

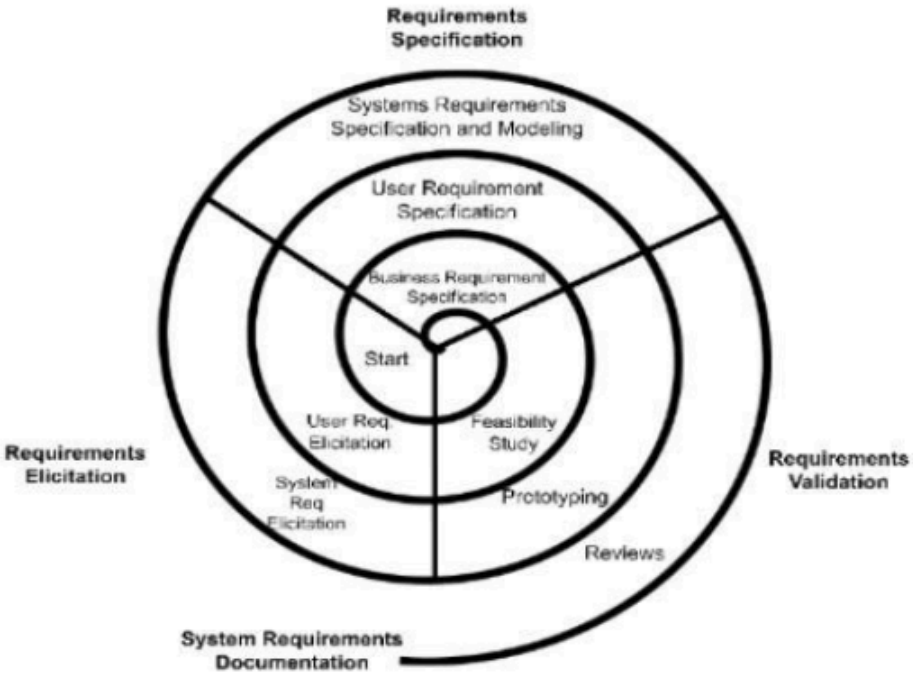


Figure-3: Spiral model

Phase-1: Planning analysis

The requirements are gathered and analyzed in the planning phase. Here user registration, searching of train, availability of seat, booking ticket and making payments are included.

Phase-2: Risk analysis phases

Here the potential risks and possible risks of IRCTC system are identified and mitigated. There are several risks such as threats of cyber security, phishing, hacking etc. The solutions such as the loading balance, techniques of encryption, and API securing is planned.

Phase-3: Engineering phases

In this phase development of the IRCTC ticket booking takes place. This system is designed with modules such as user authentication,

schedules of the train, selection of the seat and processing of the payment. The information about the ticket information and user. Information is stored in Databases

Phase-4: Evaluation phases

To validate the performances, security of the user and the usability, the developed IRCTC booking system is tested in this phase of the spiral model of SDCL models. Here feedbacks from user is taken regarding the speed of ticket searching, experience booking and success rate. New iteration begins for further enhancements such as dynamic pricing, integration of chatbot or AI-driven seat recommendations.

3.Comparision table

Feature	Waterfall model	Incremental model	Spiral model
Approach	Linear and sequential	Iterative and built in parts	Iterative along with risk analysis
Flexibility	It is rigid and no changes are allowed	Moderate flexibility	Highly flexibility
Risk handling	Low	Medium	High
Cost	Low for the small projects	Medium	High
Best for	It is small and well-defined projects	Medium-sized projects with evolving needs	Large and high-risk projects

To develop a complex system as the IRCTC Railway Ticket booking system, it is very necessary to carefully select a right Software development life cycle model. Selecting the proper and right SDLC model plays a vital role and can significantly impact the success, flexibility and also the maintainability of the system. From the above table we can clearly see that the Spiral model is best suitable for the IRCTC Train booking system. It is because the waterfall model is best used for the small data and also it is used for working of small projects. Hence due to this reason the Waterfall model fails to hold good for the IRCTC ticket booking system for train where lakhs of people use this software in a day. Similarly, the incremental model is used mainly for the medium sized data. Also, it handles medium sized project. Hence for some extent even the incremental model fails to hold good for

IRCTC ticket booking system for train. But IRCTC Train booking system deals with the large and high-risk projects. Hence the spiral model is best suitable for IRCTC Train booking system for dealing with large and high-risk projects. For a large-scale system like IRCTC Railway Ticket booking, Spiral Model is the best suited model. This model of Spiral, effectively handles and manages risk and allows the iterative development and continuous feedback is integrated.

4.Requirements Engineering :

1.Requirements document:

The IRCTC train booking system is an online software or a platform which helps user to book a ticket in train, search for the train in required time and on required day, modify the tickets or seats and even cancel the tickets of the train. It facilitates and ensures transaction securely, provide the true and real-time train availabilities of trains and offer various and numerous services like PNR checking status, cancellation of the tickets and also the special bookings of train.

1.1 Functional requirements:

a) **User Management:** The users of the IRCTC ticket booking app must be able to register or login safely and securely without the leakage of their information. There must be ability or an option for the users of the software to reset or change the password as there is a chance for the users to forget the password and if there is no option to change the password it will be a problem for the user. Also, management of the profiles must be allowed by the software. This includes updating personal details of the user and also the information related to contact of the user.

b) **Train booking and search:** Users should be able to reset their passwords. Users should be able to view train details, availability, and fare information. Users should be able to book tickets and receive a confirmation email or SMS. Users should be able to choose their preferred seats (if available)

c) **Transactions and payments**: The system should accept a number of payment methods, including credit/debit cards, UPI, net banking, and wallets. Users should receive a confirmation of successful payments. Users should be able to manage their

profiles, including updating their contact information and passwords.

d) **Ticket and PNR management**: It should be possible for users to download or print e-tickets and check their PNR status. It should be possible for users to change or cancel tickets they have reserved.

e) **Alerts and notification**: Users ought to get real-time booking status updates. When there are train cancellations, delays, or re-schedules, the system ought to notify users.

f) **Customer help and support**: Customer Support like FAQs and a help desk should be available through the system. It should be possible for users to file complaints and monitor their progress.

1.2 Non-functional requirements:

a) **Performance**: At least 10,000 users should be able to access the system at once. It should take less than three seconds to reserve a ticket.

b) **Security and safety:** Every financial transaction and user's information should be encrypted. For login, two-factor authentication (2FA) ought to be used.

c) **Availability:** The system's uptime should be 99.9%. The system should be expandable to accommodate high booking periods.

d) **Usability or practicality:** The system should adhere to accessibility guidelines (WCAG 2.0) and have an intuitive user interface that is accessible through online and mobile apps.

e) **Compliances or Adherence**: The system must abide by both government and IRCTC regulations. The system should follow data protection laws (GDPR, IT Act 2000).

2.Requirements validation:

2.1 Techniques for validation:

a. **Requirement reviews**: In IRCTC train reservation system, requirement reviews consist of examining system specifications to ensure completeness, clarity, and feasibility. Booking workflows, payment transactions, and ticketing are reviewed by st akeholders such as railway personnel, developers, and end-users. Reviews pinpoint gaps, ambiguities, or conflicting requirements prior to development. This ensures easy operation, prevents expensive rework, and matches the system to user requirements and railway regulations.

b. **Prototyping**: Prototyping in IRCTC means developing interactive models of prominent features such as ticket booking, PNR status, and cancellations. A clickable UI prototype enables stakeholders to visualize functionalities and provide a user-friendly interface. This identifies usability problems early on, streamlines workflows, and provides accessibility to various users. Prototyping assists in improving real-time seat availability display, fare calculation accuracy, and enhancing transaction speed prior to full-scale deployment.

c. Test cases: IRCTC test cases confirm key functions like train search, ticket reservation, seat availability, and payment processing. They test cases like successful payments payment failure, invalid user inputs, and load conditions during peak hours. Edge cases confirm that the system efficiently processes invalid PNRs, refunds, and Tatkal reservations. Properly designed test cases enhance reliability, reduce booking failures, and improve overall system performance.

d. **Stakeholder feedback:** Stakeholder comments for IRCTC are collected from travellers, railway officials, and

customer care groups. Comments determine usability problems, payment errors, changes in ticket confirmations. Constant feedback optimizes mobile and web platforms, load-handling capacity, and user-friendly functionality such as auto-fill options. Positive feedback ensures improved user experience, greater transparency in fare determination, and smoother ticket amendment possibilities.

e. **Traceability matrix:** An IRCTC traceability matrix ensures that all functional and the non – functional requirements are traced to related test cases. It monitors features such as updates in train schedules, dynamic pricing, and refund policies, making sure all are properly tested. The matrix identifies missing validations, ensures government policy compliance, and preserves system integrity. It also aids in impact analysis when changes are made.

2.2 Potential challenges:

a. **Ambiguity in user requirements:** Ambiguity in IRCTC requirements occurs when users anticipate different fare calculations, ambiguous refund rules, or perplexing seat availability displays. Misunderstandings lead to improper bookings, delay, or user dissatisfaction.Clearly specifying requirements like fare disb ursements and ticket cancellation policies facilitates user trust. User-friendly interfaces, exhaustive FAQs, and interactive customer support alleviate confusion and enhance experience.

b. **Scalability issue:** Scalability issues in IRCTC happen during peak seasons like festival seasons or Tatkal booking timings when millions of users attempt to book at the same time. Heavy traffic can lead to server slowdowns or crashes. Use of cloud-based architecture, load balancing, and database query optimization ensures smooth performance. Str ong caching mechanism and predictive ticket demand analysis ensure additional scalability.

c. **Secure concerns:** IRCTC security threats consist of payment scams, phishing, and data breaches of personal

information. Hackers can try unauthorized access to the user's account or fake ticket availability. Utilizing multi-factor authentication (MFA), SSL encryption, and secure payment gateway reduces threats. Periodic security scans, CAPTCHA, and artificial intelligence (AI)-based fraud detection further strengthen the system to ensure secure transactions and confidentiality of users.

d. **Integration with existing systems:** Integrating IRCTC with railway management, payment gateways, and government databases is intricate. Real-time seat availability syncing with train schedules demands seamless data exchange. Problems occur when legacy railway systems do not sync with contemporary APIs. Effective API design, middleware solutions, and periodic testing ensure seamless integration, avoiding ticket booking, cancellation, and real-time update errors.
e. **Regulatory compliances:** IRCTC has to adhere to government regulations such as GST implementation, data protection legislations, and railway policies. Failure to do so can attract legal consequences or service disruptions. Compliance with digital payment standards, protection of personal data (according to IT Act), and accessibility standards is paramount. Auditing regularly, encryption policies, and conformity with railway ministry guidelines enable legal compliance and system stability.

5.Conclusion:

In the conclusion, the IRCTC ticket booking system exemplifies the capability of digital solutions to enhance convenience, decrease administrative overhead, and facilitate enhanced travel planning for millions of users. As it undergoes frequent updates and advances in technology, it keeps developing, making it an easy-to-use and efficient ticket booking platform in India. The project exemplifies the need for digital transformation in bringing modernization to traditional services and addressing the rising needs of an increasingly mobile and technology-embracing population.