Assignment Solution

# Report information

* Student name: Obada Al-halaybeh
* Student ID: 23110107
* Student Section number: 1
* Project Title: Around the world

# Table of Contents

[Report information 1](#_Toc183207137)

[Table of Contents 1](#_Toc183207138)

[Part 1: Software Lifecycle Models 3](#_Toc183207139)

[Describe two iterative and two sequential software lifecycle models. 3](#_Toc183207140)

[Explain risk Management: Definition, Steps, and Risk Strategies in Agile and Waterfall Models 3](#_Toc183207141)

[Explain the purpose of a feasibility report in software development and discuss the components of a feasibility report. 3](#_Toc183207142)

[Describe criteria and methods for comparing two technical solutions. 3](#_Toc183207143)

[Part 2: Software Investigation and Analysis 3](#_Toc183207144)

[Conduct a software investigation to understand the requirements for the tourism booking software by gathering requirements using two techniques (with evidence), analyzing current systems, and identifying key functionalities needed. 3](#_Toc183207145)

[Create a feasibility report for the tourism booking software system. 3](#_Toc183207146)

[Create a Software Requirements Specification (SRS) document for the tourism booking software system. 3](#_Toc183207147)

[Discuss why you might select a particular lifecycle model for developing a tourism booking software system. 4](#_Toc183207148)

[Part 3: Software Design Techniques and Behavioural Tools 4](#_Toc183207149)

[Discuss using two examples the suitability of software behavioural design techniques. 4](#_Toc183207150)

[What is the difference between a finite state machine (FSM) and an extended FSM, and examples of where each might be used in a tourism booking software system? 4](#_Toc183207151)

[Create a Software Design Specification (SDS) document for the tourism booking software system. You must provide two techniques; one of the techniques must be a Data Flow Diagram (DFD) in two forms: context level, level 0. 4](#_Toc183207152)

[Provide an implementation of the tourism booking software system. You can use tools like Adalo, WordPress. You may use other tools with the approval of your instructor. The implementation must cover following requirements (user registration, package creation, and reservation management) you must provide screenshots for each feature in addition to your implementation URL or Code. 4](#_Toc183207153)

[Part 4: Software Quality and Testing 4](#_Toc183207154)

[Discuss two approaches to improving software quality. 4](#_Toc183207155)

[Evaluate the process of undertaking a systems investigation that you use in Part 2.1 with regard to its effectiveness in improving software quality. 4](#_Toc183207156)

[Create a Requirements Traceability Matrix (RTM) document for the tourism booking software (The RTM must be relevant to your implementation in Part 3.4 minimum five functional requirements with two test cases for each one) . 4](#_Toc183207157)

[Part 5: Evaluation and Justification 5](#_Toc183207158)

[Assess the merits of applying the Waterfall lifecycle model to a large software development project. 5](#_Toc183207159)

[Evaluate how criteria like cost, time, resources, and legal requirements affect the decision-making process and the overall feasibility of the tourism booking software project. 5](#_Toc183207160)

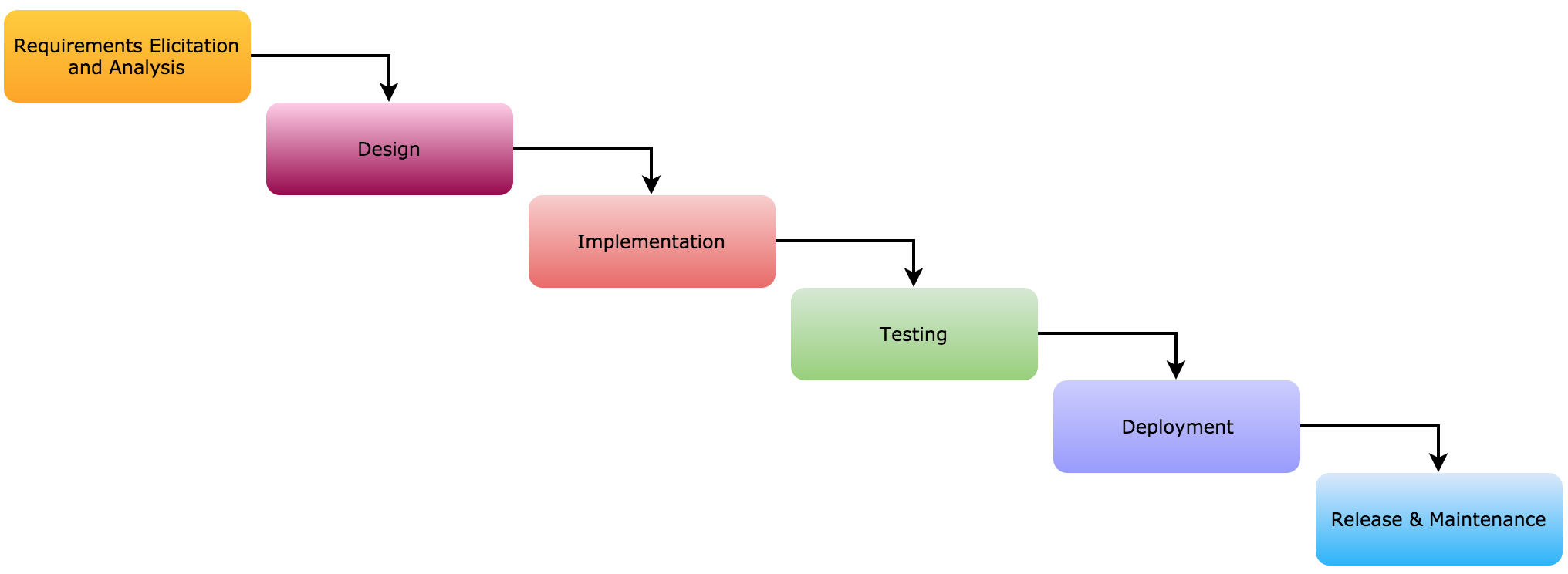
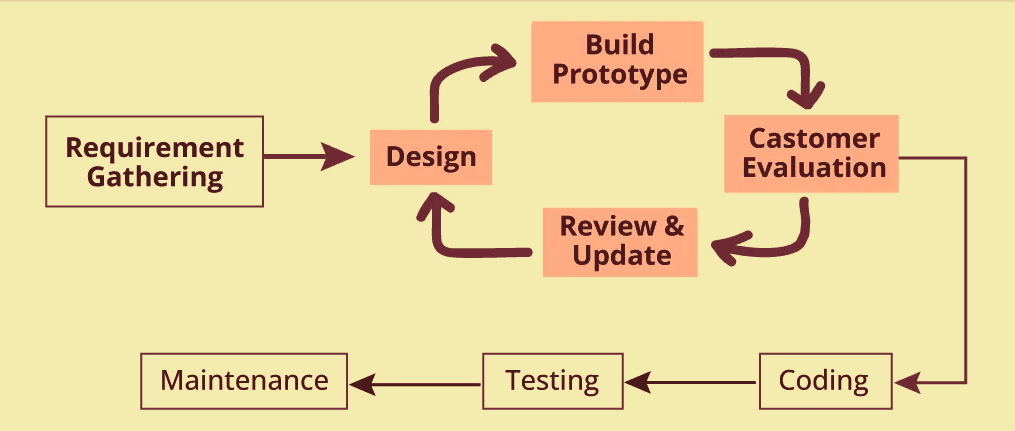
[Justify the use of data analytics, machine learning, and other data-driven approaches to enhance the performance, reliability, and user satisfaction of the tourism booking software. 5](#_Toc183207161)

[References 5](#_Toc183207162)

# Part 1: Software Lifecycle Models

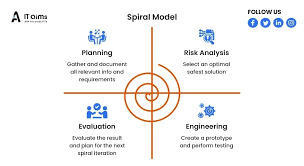
### Describe two iterative and two sequential software lifecycle models.

1.sequential

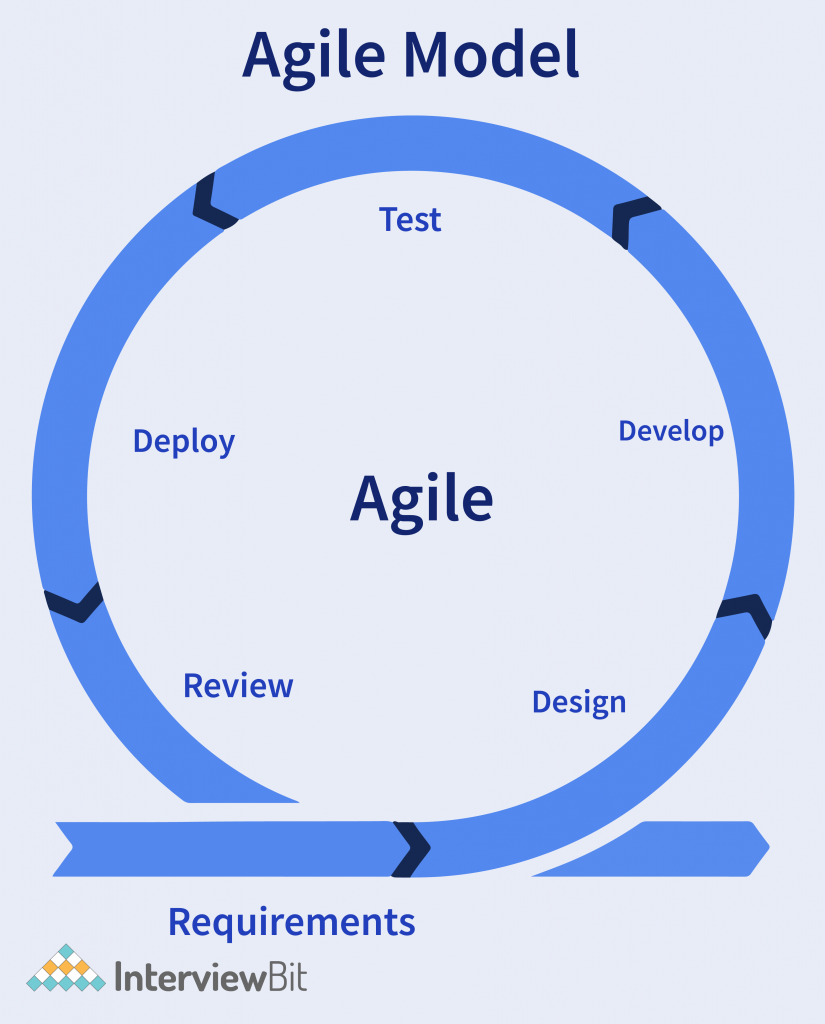
* Waterfall model
  + The waterfall model is one of the first models to be introduced it is linear and structured approach, in this model the phase must be fully completed before going to the next one.
  + It has 6 phases: requirement, design, implementation, testing, deployment, maintenance.
  + There is some advantges for the waterfall like:
    - Easy to understand
    - Provides structure to inexperienced staff
    - Milestones are well understood
    - Sets requirements stability
    - Good for management control
    - Easy to manage
  + There is some disadvantges for the waterfall like:
    - All requirements must be known upfront
    - Error can be fixed only during the phase
    - Can give a false impression of progress
    - Testing period comes quite late in the developmental process
    - Inflexible to changes after the initial phase
  + We use waterfall method when the requirements are well known, technology is understood, product definition is stable, when there is new version of existing product or porting an existing product to a new platform.
* Prototyping model
  + Prototyping model is one of the most popular used SDLC model, this model focus on building working prototype early then refine it based on customer feedback to get the final product.
  + The phases of prototyping model: requirement gathering, prototype development, feedback and refinement, development, deployment and Maintenance.
  + There is some advantges for the prototyping model like:
    - Users are actively involved in the development
    - Errors can be detected much earlier
    - Missing functionality can be identified easily
    - Reduces misunderstandings about requirements
  + There is some disadvantges for the prototyping model like:
    - Risk of over-reliance on the prototype, leading to delays
    - The cost of developing a prototype is a total waste as the prototype is ultimately thrown away
    - May not work well for systems with strict timelines.
  + There is 4 types of prototyping models: Rapid Throwaway prototypes, Evolutionary prototype, Incremental prototype, Extreme prototype
  + We use prototyping method when the desired system needs to have a lot of interaction with the end users, on online systems and web interfaces.

2. Iterative

* Spiral Model
  + spiral model is one of the most important SDLC models, that provide support for risk handling, in its diagrammatic representation, it looks like a spiral with many loops.
  + The number of loop is unknown and can vary from project to project, each loop is considered a phase of the software development process, the project manager can set the number of phases needed based on the project risks.
  + The radius spiral at any point represents the cots of the project so far and the angular represents the progress mad so far in the project the spiral model includes iterative nature of the prototyping model and the linear nature of the waterfall model.
  + The phases of spiral model: planning, risk analysis, engineering, evaluation.



* + There is some advantges for the spiral model like:
    - High amount of risk analysis to avoid risk
    - Good for large and mission-critical projects
    - Software is made early which allow the customer to see it fast
  + There is some disadvantges for the spiral model like:
    - High cost
    - Not good for small projects
    - Risk analysis requires highly specific expertise
  + spiral model is used when the costs and risk evaluation is important, also for long term projects, when the user is not suer of his needs.
* Agile Model
  + Agile model is a methodology for modeling and documenting software system based on best practices, it combine both iterative and incremental development with the focus on process adaptability and customer satisfaction by rapid delivery.
  + Agile methods brake the product into small incremental builds provided in iterations each iteration take form one to three weeks, this methodology is more flexible than the traditional ones making it better for fast changing environment.
  + The phases of agile model: Requirements gathering, design, develop, test, deploy, feedback



* + There is some advantges for the agile model like:
    - Customer satisfaction by rapid, continuous delivery of useful software
    - People and interactions are emphasized rather than process and tools
    - Highly flexible and adaptive to changes
    - Working software is delivered frequently
  + There is some disadvantges for the agile model like”
    - In case of some software deliverables, especially the large ones, it is difficult to assess the effort required at the beginning of the software development life cycle
    - There is lack of emphasis on necessary designing and documentation
    - Requires constant communication and collaboration

### Explain risk Management: Definition, Steps, and Risk Strategies in Agile and Waterfall Models

A risk is a problem that has not yet happened, it can cause some loss or threaten the progress of the project. And we could manage it by eliminating the problems be for it has a negative impact on the project.

The way of managing a risk can be done by identifying the potential problem through brainstorming, creating a risk list depends on the risk that occurred repeatedly in other projects. We should also do a risk assessment following some steps like looking for the issues that cause the risk in the project, determining the impact of the problem and the likelihood of it occurring lastly we could assign probability and impact values ranging from 1 to 10.

The risk management has a lot of strategies that could help us manage the risks:

* Risk avoidance and mitigation  
  this technique aims to eliminate the occurrence of the risks entirely, to avoid the risks we should reduce the scope of the project by removing non-essential requirements.
* Risk transfer  
  this technique is used when the project is too large to be handled by one team and can’t be split up between more than one team, so the company search for another outside company that can take a portion of the project.
* Risk acceptance  
  this technique make you take the risk in order to complete the project, as it make since to take a level of risk so you have some time to find out what needs to be done and how long it will take. Note this strategy may contain some negative effects cause a delay on the deadlines for the project.
* Risk monitoring  
  in this strategy the risk should be watched continuously, and we should reevaluate the risk, the impact of it and the probability of it occurring. This way we make suer that the risk dangers have been discovered and rediced.
* **Risk Strategies in Agile**

The risk management here take short time frames, the risk table is reevaluated at every sprint planning meeting, risks are also discussed at every stand up meeting, also at the end of each iteration, in conclusion using agile reduce the risk related to budget.

* **Risk strategies in Waterfall**

In waterfall you need to plan for the risk preparation beforehand the estimation about the likelihood and severity of a risk can be not accurate all the time, it is important to define roles and responsibilities for constant monitoring and controlling risks,overlapping stages of development and long processes are all sources of risk in waterfall environment.

|  |  |  |
| --- | --- | --- |
| Aspect | Agile | Waterfall |
| Approach | Iterative and adaptive | Sequential and structured |
| Risk Identification | Continuous (before and during each sprint) | Early (during planning and design phases) |
| Mitigation Strategy | Frequent deliverables and stakeholder feedback | Detailed plans and phase-gate reviews |
| Response to Changes | Flexible, with ongoing adaptation | Rigid; requires formal change control process |
| Risk Visibility | High throughout the project | Often hidden until later stages |

### Explain the purpose of a feasibility report in software development and discuss the components of a feasibility report.

Feasibility report is the result of the software development team’s feasibility analysis, it includes recommendations on how the software development should be continued in the future by the software development team, it also could include information about changes to the software scope, technical skills budget, and schedule. The feasibility report aim to determine the chances of the project succeeding and its done by doing a detailed analysis that considers all the critical aspects of the project.

**Components of the feasibility report**

* Legal feasibility  
  In legal feasibility the project is analysed to be compatible with the legal requirements in most of the world, that includes data protection acts, social media laws and copyright.   
  in conclusion the legal feasibility shows whether the project meat the legal requirement or not.
* economic feasibility  
   economic feasibility is a process of identifying the financial benefits and costs that is related to the project development. This study includes the costs of hardware and software, labour costs and operational costs this is the tangible costs for the intangible costs it could be loss of customer goodwill, employee morale and operational inefficiency over all economic feasibility analyse whether the project will be profitable or not.
* technical feasibility  
  technical feasibility evaluates if the current resources like the hardware and software along with the technical requirements are good to create the project, it also gives a report whether there exist required resources and new technologies which will be used for proposed software development, as well as the study check the technical skills of the software development team, the viability of using current technologies and how hard it is to maintaining and upgrading the technology.
* operational feasibility  
  operational feasibility will evaluate how the project will have effect on the company processes, what procedures should be implemented and what is the effort should be made to keep the company running. It also looks at the services that is provided and if the project algin with the workflow and user needs.
* scheduling feasibility  
  scheduling feasibility looks on the time flow, it study the timeline for the project and analyse it to give estimate deadlines for the task and project, this study is important since it can determine if the project will be done on the time or not.

### Describe criteria and methods for comparing two technical solutions.

For comparing two technical solutions there are some criteria and methods need to be present.

For the criteria we have cost, security and customizability.

**cost**

Cost refers to the overall financial investment required for development, deployment, and ongoing maintenance, a less expensive option appears more appealing, but functional needs shouldn't be sacrificed for that.

**customizability**

Customizability is the level at which the system is able to satisfy the user or the organization interface requirements. Highly customizable systems can be set for specifics workflows, features and even user interfaces.

**Security**

Security measures assess how well the system limits access to user data and guarantees that all legitimate user data is correct and trustworthy. Most of these security measures are crucial for systems that handle private or sensitive data.

For the methods we have ROI (Return on Investment) analysis, Cost-benefit analysis and Risk analysis

**ROI (Return on Investment) analysis**

The ROI (Return on Investment) analysis is the cash flow comparison that occurs between the costs and the returns of the investments. Such cash flow comparison aids in answering the crucial question: Do the benefits outweigh the expenses in the first and subsequent stages?

**Cost-benefit analysis**

According to Benson's cost-benefit analysis, it is helpful to weigh the costs associated with fixing a given problem against the benefits that result from doing so in order to determine whether or not it is worthwhile.

**Risk analysis**

Risk analysis deals with the assessment of all perceivable potential risks that may arise from the implementing of the solution and how likely or impactful they are. This assists in appreciating the uncertainties and preparing the measures to reduce them.

For example we have Facebook and LinkedIn

Cost:

The costs of Facebook Operations increase largely due to its global networking and content requirements. Due to having a professional audience LinkedIn is also able to reduce a large chunk of Operating costs because of having earnings to cover most of the services rendered.

Customizability:

Advertising features offered by Facebook are a limited form of customisation for businesses whereas for individual users they provide no such features at all. LinkedIn offers more freedom to users and the companies themselves regarding the business profile visibility, content visibility, recruitment tools all which are tailored according to specific requirements of the user and the companies.

Security:

As a social media giant, Facebook has been in the spotlight for several regulatory issues related to privacy. Still Facebook offers the proper security which many people wish for, for example it offers two-factor authentication. Although LinkedIn revolves more around business data and information, it has had security breaches but has made it an objective to protect people’s information and be GDPR compliant.

In terms of return on investment (ROI), Facebook's user growth, engagement, and advertising strategy outweigh its overall operating costs. Although LinkedIn does not benefit from the same degree of economies of scale, its professional networking niche market and premium subscription-based business model provide a return on investment.

From cost-benefit ratio perspective, it appears that Facebook has a larger audience which justifies the revenue from advertising thus explaining the cost incurred. In the case of LinkedIn, the benefits are the focus on professionals thus being useful to businesses and users who wish to network or enhance their careers, thus allowing it to have less costs.

For Facebook Risk analysis has focus on privacy and exposure of critical data whereas in case of LinkedIn the risk is more towards the preservation and protection of professional data and user information.

# Part 2: Software Investigation and Analysis

### Conduct a software investigation to understand the requirements for the tourism booking software by gathering requirements using two techniques (with evidence), analyzing current systems, and identifying key functionalities needed.

1. Method 1: survey  
   **evidence**   
   A screenshot of a computer

   Description automatically generated

A survey was conducted to gather feedback and preferences about travel booking platforms. the survey had questions that focused on preferences, challenges faced, wanted features and payment methods.

For the findings it will be shown in the images I will provide for the answers.

A pie chart with numbers and a few words

Description automatically generated with medium confidence

A graph with purple bars

Description automatically generatedA graph with purple bars

Description automatically generated with medium confidence

A pie chart with text below

Description automatically generated

A pie chart with text

Description automatically generated A pie chart with numbers and a few words

Description automatically generated with medium confidence A graph with purple bars

Description automatically generated with medium confidence A pie chart with numbers and a triangle

Description automatically generated A graph with purple bars

Description automatically generated with medium confidence

A screenshot of a chat

Description automatically generated

A screenshot of a website

Description automatically generated

1. Method 2 background study

I made a background study on **Expedia** a big booking platform and took user reviews from **trustpilot,** to look up strengths, weaknesses, and user feedback on the existing system. This study gave me an understanding look at the industry standards and gaps that need to be filled with the new app.

The findings.

Strengths, it is a comprehensive platform as it combines flights, hotels, and transportation booking all in one platform, it also has a reward program that allow users to earn points when using the app to book.

Weaknesses, the customer support was most of the time late to response and solve the problem, there was hidden fees on check out and they had a major problem on refunding.

1. Requirements gathered.

Based on the survey and background study, the following requirements have been identified:

1.functional requirements

* User registration and profile management: allow the user to create account and mange it as he pleases
* Search and booking: allow the user to look for the travel options with filters and preferences based on his history.
* Payment processing: include versatile payment methods like credit card, payPall and mobile wallets.
* Reward program: give the user loyalty points when he use the app and let him exchange it for discounts.
* Customer support: offer more than one way to reach out to us for help.

2. Non-Functional Requirements

* Performance: make sure the system can handle any type of pressure.
* Security: have strong data encryption and comply with GDPR standards.
* Availability: the app should be up at all the time as much as possible.

### Create a feasibility report for the tourism booking software system.

1. Introduction:

The (app name) aim to provide the users with a platform to search and book travel services like flight and hotels. This system will focus on addressing some of the problems that face the users when trying to use other apps.

This report is done to study the all the aspects of the project to see if its going to fail or succeed.

1. Legal Feasibility:

* The legal feasibility ensures that the app complies with regulations, protects user right, and mitigates potential risks.
* (app name) must follow the data protection and privacy regulations like GDPR(general data protection regulation) that state the software must protect the user data and ensure lawful processing and the user must consent before getting there data collected. CCPA(California privacy act) this is applicable for users in California. The software must also comply with the local privacy laws like the LGPD in Brazil and PIPEDA in Canada.
* Payment compliance, PCI DSS (Payment Card Industry Data Security Standards) all the payment should be adhered to PCI DSS to make sure of the credit card security and this requires encrypted data transactions and vulnerability testing.
* International trade laws, the software should comply with laws cross-border services like the different tax if the user book from different countries. Legal disclaimer we should include disclamers to limit the liability for some of the issues like if the airline contracted with the app got delays.
* In conclusion, the legal feasibility study provided the system adheres to global and local laws

1. Economic Feasibility:

* economic feasibility is done to determine if the project can be financially viable in both costs and benefits.
* one-time cost : (app name) will be developed to be suitable for mobile and this going to divide the development process into designing, coding, testing and implementation, this will cost about $20,000 and we have the marketing costs to promote for the app launch and this will cost about $5,000, over all the total of the one time costs will sum up to a $25,000.
* Recuring costs: this cost will be summed up annually like the maintenance costs from bug fixing, system updates and performance optimization, those will happen occasionally so this will be around $3,000/year, and we have the hosting fees that will be about $6,000/year.
* Tangible benefits: are going to be from booking fees as the user will pay the fees from the transaction for flights and hotels this going to be around $60,000/year, and the advertisement and the partnerships will generate about $5,000/year so the annual revenue will be $65,000.
* Intangible benefits: are benefits that can’t be measured with money but can get a hug value to the project like the customer satisfaction from the enhanced experience he will get.
* ROI(return on investment)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Year 0 | Year 1 | Year 2 | Year 3 | total |
| Total Benefits | 0 | 65,000 | 65,000 | 65,000 | 195,000 |
| Total Costs | 25,000 | 9,000 | 9,000 | 9,000 | 52,000 |
| Net Benefits total benefits – total costs | -25,000 | 56,000 | 56,000 | 56,000 | 143,000 |

ROI = (Total Benefits – Total Costs) / Total Costs

ROI = 143,000/52,000 = **275%**

1. Technical Feasibility:

* Technical feasibility assesses the resources, technology, and expertise available to implement the project.
* Technological stacks: frontend, backend, database, and the host cloud we need to make suer that the developing tools used are up to date in the market so that we can have all the new features.
* Technical team skills are very important to the project as it will cut handing out time as well as ensure the security of the project and user data.
* In conclusion, the technical feasibility is done to reduce the potential risk that could affect the app.

1. Operations Feasibility:

* Operational feasibility assesses how the system will integrate into daily operations and meet user expectations.
* The system should preform as expected on a high demand period like the summer vacation, wither it’s a customer support issue or reserving issue, as for the customer support there will be ai powered bots that can answer simple questions, for the reserving the system should not make any delay that may affect the reservation in any way possible.

1. Scheduling Feasibility:

The scheduling feasibility evaluate wither the project can be completed within the proposed time or not

A screen shot of a chart

Description automatically generated

* The total time is 186 days.
* All the teams should be trained to be able to finish the task before the deadline to avoid latency.
* The task should be handed before one week from its deadline.

1. Conclusion:

* Travel booking software has legal, economic, technical, operational and temporary feasible. The system will deal with the major issues found during the audit. Complies with legal and technical standards and create a competitive advantage in the travel booking market.

### Create a Software Requirements Specification (SRS) document for the tourism booking software system.

**Software Requirements Specification (SRS)**

**1. Introduction**

**1.1 Purpose**  
This SRS document describes the functional and non-functional requirements for the development of a **Tourism Booking Software System**. The system will provide users with an intuitive platform to book flights, hotels, and transportation while offering personalized recommendations, reward programs, and seamless customer support.

**1.2 Scope**  
The Tourism Booking Software will support:

* A user-friendly interface for searching and booking travel services.
* Secure payment methods, including credit cards, PayPal, and mobile payments.

This system aims to address challenges such as customer support delays, hidden fees, and refund issues.

**1.3 Intended Audience**  
This document is intended for:

* **Development Team**: To understand the system requirements.
* **Project Stakeholders**: For review and feedback on the system’s functionality.

**1.4 Intended Use**  
The document will guide:

* System design and development processes.
* Cost estimation and timeline planning.
* Risk assessment and mitigation strategies.
* Road map development and features back log.
* Testing Acceptance criteria .

**1.5 Definitions and Acronyms**

* **SRS**: Software Requirements Specification
* **GDPR**: General Data Protection Regulation

**2. Overall Description**

**2.1 Product Perspective**  
The Tourism Booking Software will serve as a centralized platform for managing travel bookings. It will replace manual processes and improve user experience by automating tasks like flight searches, hotel reservations, and payment processing.

**2.2 Product Features**

* Search and booking for flights, hotels, and transportation.
* Reward programs for frequent users.
* Personalized recommendations based on user preferences.

**2.3 User Classes and Characteristics**

* **End Users**: Travelers who book travel services.
* **Admins**: Manage the system, including travel package updates and support requests.

**2.4 Operating Environment**  
The system will operate on:

* Mobile platforms (Android and iOS).

**3. System Requirements**

**3.1 Hardware Requirements**

* **mobile-compatible infrastructure.**

**3.2 Software Requirements**

* **Frontend: React.js or Angular for user interface.**
* **Backend: Node.js or Python for server-side logic.**

**3.3 Integration Requirements**

* **Payment gateway integrations like PayPal.**

**3.4 Regulatory Requirements**

* **Compliance with GDPR for data protection.**

**3. Functional Requirements**

1. **User Registration and Profile Management**:
   * Users can create accounts.
   * Users can log in.
   * Users can manage their profiles.
2. **Search and Booking**:
   * Users can search for travel options using filters like location.
3. **Payment Processing**:
   * Support for secure payments via credit cards, PayPal, and mobile wallets.
4. **Reward Program**:
   * Points-based loyalty system for frequent bookings.
5. **Customer Support**:
   * Multi-channel support (phone, email, live chat, and social media).

**4. Non-Functional Requirements**

1. **Performance**:
   * The system should handle up to 1,000 concurrent users.
2. **Security**:
   * SSL encryption for data transmission.
   * Compliance with GDPR standards.
   * Compliance with PCI DSS.
3. **Availability**:
   * Maintain system uptime at an acceptable level.

**5. User Stories (user requirments)**

* **As a user**, I want to see personalized travel recommendations so I can quickly find options that suit me.
* **As a user**, I want to pay securely using my preferred method to avoid payment issues.
* **As a user,** I want to be able to search and filter available travel options so I can find the best options for my situation.
* **As a user,** I want to be able to update my profile.
* **As a user,** I want to be able to get support when needed so that I can resolve any issues I face in a timely manner.
* **As an admin,** I want to be able to reward the users based on loyalty and frequency so that I can maintain a healthy number of users.
* **As an admin**, I want to manage travel packages to ensure accurate availability for users.

**6. Challenges and Mitigation Strategies**

1. **Hidden Fees**:
   * Mitigation: Ensure all charges are displayed upfront during booking.
2. **Refund Issues**:
   * Mitigation: Automate the refund process and communicate clear timelines to users.

### Discuss why you might select a particular lifecycle model for developing a tourism booking software system.

Choosing the right software development lifecycle (SDLC) model is essential for ensuring the success of a tourism booking software system. The selection of a lifecycle model is influenced by several factors, including project size, complexity, stability of requirements, and the necessity for flexibility. In this text, I will explain why the Agile model is an ideal option for creating a tourism booking software system.

* **Flexibility and Adaptability**:
  + Tourism booking systems typically need regular updates and modifications in response to user input, market developments, and new regulations. The Agile model offers great flexibility and  enables ongoing iteration and enhancement. This is especially beneficial in a dynamic field such as tourism, where consumer preferences and market circumstances can shift quickly.
  + For instance, when users ask for a new feature, such as adding a payment option (cryptocurrency), the Agile model enables the development team to implement this adjustment in the next iteration without interfering with the overall project.
* **Customer-Centric Approach**:
  + Agile prioritizes providing value to the customer by frequently releasing functional software. In the realm of a tourism booking platform, this implies that users can begin reaping the advantages of the system early on, even if certain features are not completely finalized. This cyclical method guarantees that the software develops according to genuine user input, resulting in increased customer satisfaction.
  + For example, the system could start with essential functions such as booking flights and hotels, and subsequently incorporate more sophisticated features like tailored travel suggestions or reward programs.
* **Risk Management**:
  + The iterative characteristic of Agile enables the early detection and management of risks. By dividing the project into smaller sprints, the team can tackle possible problems (performance bottlenecks, security flaws) early on in the development cycle. This minimizes the chances of significant issues emerging later in the project.
  + For instance, if a vulnerability is identified in the initial sprint, it can be resolved right away, stopping it from escalating into a bigger problem in later stages.
* **Scalability:**
  + Tourism booking systems frequently require scaling to handle an increasing number of users, particularly during high travel periods. Agile’s step-by

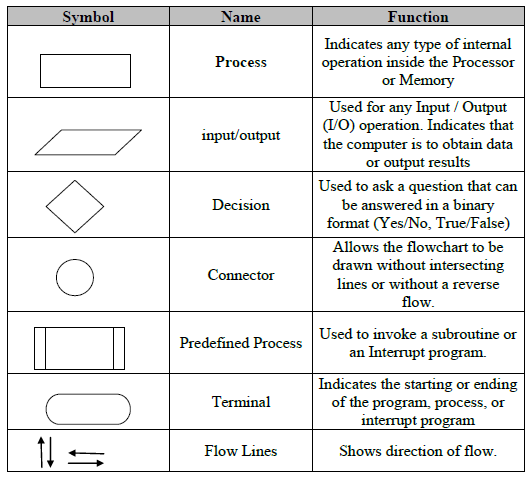
step method enables the system to expand gradually, making sure that performance and reliability are upheld as the user base increases.

* + For example, the platform may begin with a limited user base and progressively incorporate additional servers and enhance the database as user count rises.

# Part 3: Software Design Techniques and Behavioural Tools

### Discuss using two examples the suitability of software behavioural design techniques.

Techniques for software behavioural design are crucial for simulating how a system reacts to different inputs or events. These methods assist in comprehending, creating, and executing the dynamic features of a system. In the following section, I will examine two instances of behavioural design methods—Finite State Machines (FSM) and Flowcharts—and their effectiveness within a tourism booking software system.

* Finite State Machines (FSM)
  + A Finite State Machine (FSM) is a mathematical model used to represent the behaviour of a system that can be in one of a finite number of states at any given time. The system transitions from one state to another based on inputs or events.
  + In the context of a tourism booking system, an FSM can be used to model the booking process, where the system transitions through states such as "Searching," "Selecting," "Booking," "Payment," and "Confirmation."
  + Finite State Machines (FSM) are particularly effective for representing processes with distinct state changes, like the booking process in a tourism booking software application. FSMs enable the system to progress through distinct states such as "Searching," "Selecting," "Booking," "Payment," and "Confirmation," thereby guaranteeing a logical and error-free series of actions. This method is especially beneficial for processes where the system's action relies on the existing state and particular inputs, making it perfect for managing user interactions such as flight selection and payment processing.
  + Example:
    - In the tourism booking system, an FSM can model the booking process. For instance, the system starts in the "Searching" state, transitions to "Selecting" when the user chooses a flight, moves to "Booking" when the user confirms the selection, and finally transitions to "Payment" and "Confirmation" states based on user actions. This ensures that the system follows a clear and logical sequence of steps, preventing errors like attempting to proceed to payment without selecting a flight.
* Flowchart
  + A Flowchart is a graphical representation of a process or algorithm that uses various shapes (e.g., rectangles, diamonds, arrows) to represent steps, decisions, and the flow of control in a system. In a tourism booking system, flowcharts can be used to model complex processes such as the payment process, user registration, or even the overall workflow of the system. They provide a clear and visual representation of the system's logic, making it easier to handle scenarios like applying discounts, validating payment methods, or managing errors, ensuring that all possible paths and outcomes are considered for a robust and user-friendly system.  
    
  + Flowcharts are excellent for visualizing complex workflows and decision-making processes in a tourism booking system. They are particularly useful for modeling processes that involve multiple decision points, such as payment processing or user registration. Flowcharts provide a clear and visual representation of the system's workflow, making it easier to handle scenarios like applying discounts, validating payment methods, or managing errors. This technique ensures that all possible paths and outcomes are considered, leading to a more robust and user-friendly system.
  + Example:
    - A flowchart can represent the payment procedure in the tourism booking system. For instance, the system begins by verifying the user's payment method (credit card, PayPal). If the payment method is valid, the system utilizes any eligible discounts or loyalty points. If the payment is unsuccessful, the system prompts the user to attempt again or select an alternative payment method. Decision points like "Is the payment legitimate?" or "Are there sufficient loyalty points?" steer the process flow, making sure that every potential scenario is addressed appropriately.

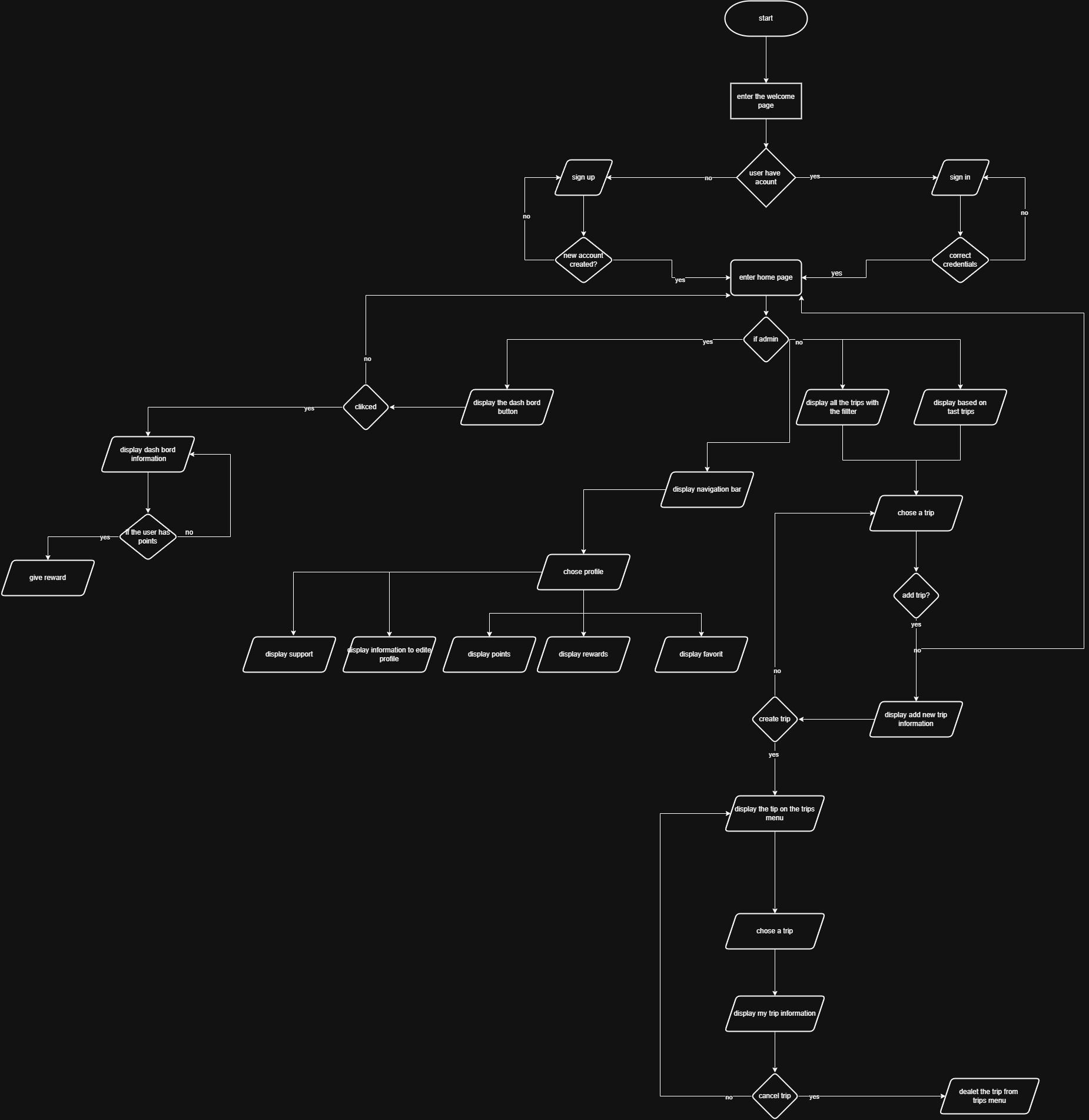
### What is the difference between a finite state machine (FSM) and an extended FSM, and examples of where each might be used in a tourism booking software system?

Here I will show 2 differences between the FSM and EFSM:

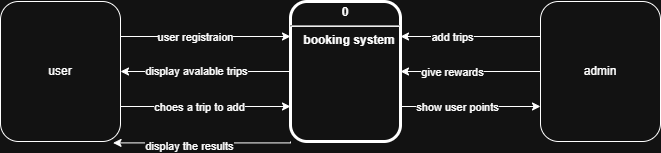
* Variables and data handling
  + For the FSM it does not support support variables or data storage. It operates purely based on states and transitions triggered by inputs or events. For example, in a tourism booking system, an FSM can transition from "Searching" to "Selecting" based on user input, but it cannot store or manipulate data like user loyalty points or payment details.
  + For the EFSM incorporate variables and data storage. This allows the system to store and manipulate data (user loyalty points, payment methods) and make decisions based on conditions. For example, in the same tourism booking system, an EFSM can check if a user has enough loyalty points to apply a discount before transitioning to the "Payment" state.
* Conditional Transitions
  + In the FSM transitions between states are triggered solely by inputs or events, without any conditions. For example, the transition from "Booking" to "Payment" occurs as soon as the user confirms the booking, regardless of any additional factors.
  + For the EFSM it allows for conditional transitions, meaning that a transition can only occur if certain conditions are met. For example, the transition from "Booking" to "Payment" in a tourism booking system may depend on conditions such as the availability of seats or the user's payment method. If the user selects "Pay with Points," the system checks if the user has enough points before allowing the transition.
* Exampls where each might be used:
  + FSM: an FSM can be used to model the basic booking process, where the system transitions through states like "Searching," "Selecting," "Booking," "Payment," and "Confirmation." Each state represents a clear stage in the booking process, and transitions are triggered by user actions.
  + EFSM: an EFSM is more suitable for modeling complex processes like payment and applying discounts. The system can use variables (user loyalty points) and conditions (payment method) to determine the next state.

### Create a Software Design Specification (SDS) document for the tourism booking software system. You must provide two techniques; one of the techniques must be a Data Flow Diagram (DFD) in two forms: context level, level 0.

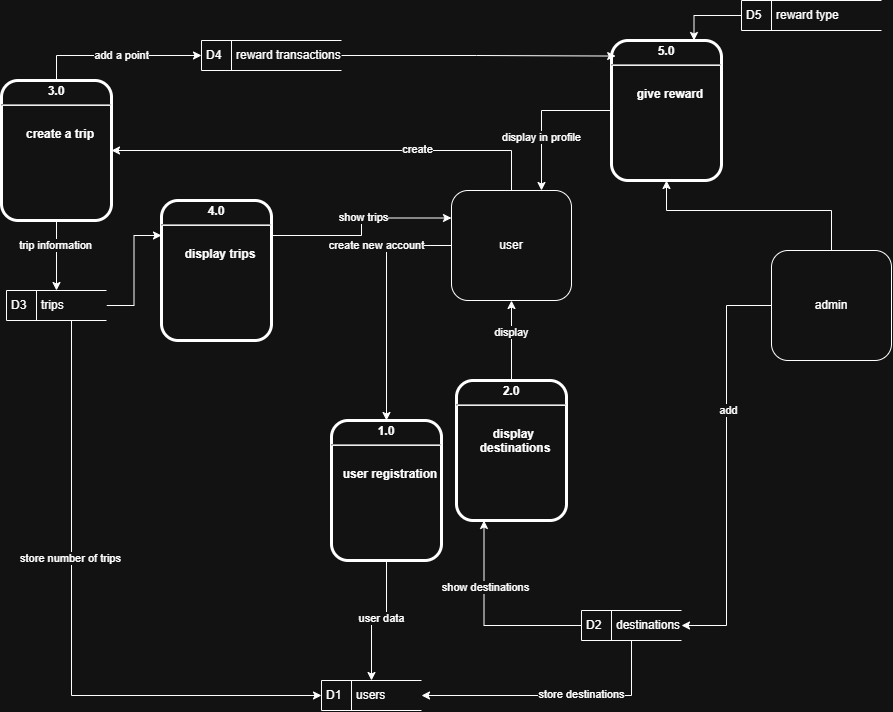
**Flowchart:**



**DFD context level**



**DFD level 0**



### Provide an implementation of the tourism booking software system. You can use tools like Adalo, WordPress. You may use other tools with the approval of your instructor. The implementation must cover following requirements (user registration, package creation, and reservation management) you must provide screenshots for each feature in addition to your implementation URL or Code.

<https://obada-alhalaybehs-team.adalo.com/around-the-world>

A screen shot of a phone

Description automatically generatedA screenshot of a login screen

Description automatically generatedA screenshot of a login form

Description automatically generatedA screenshot of a phone

Description automatically generated

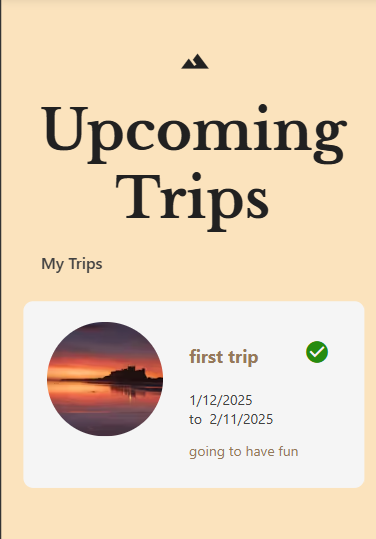
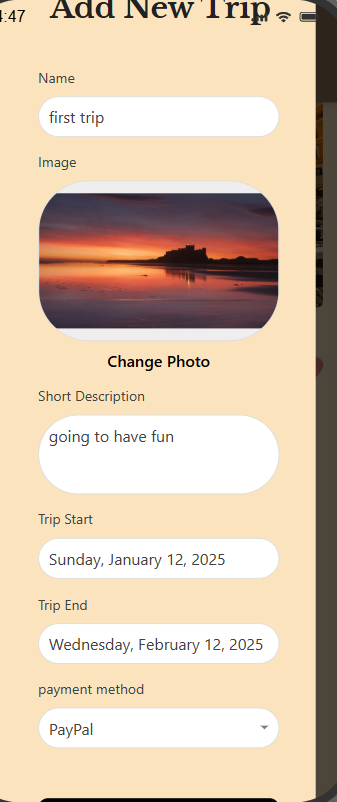


A screenshot of a computer

Description automatically generatedA screenshot of a hotel

Description automatically generatedA screenshot of a phone

Description automatically generatedA white paper with black text

Description automatically generated

A screenshot of a phone

Description automatically generated

A screenshot of a phone

Description automatically generated

A screenshot of a cell phone

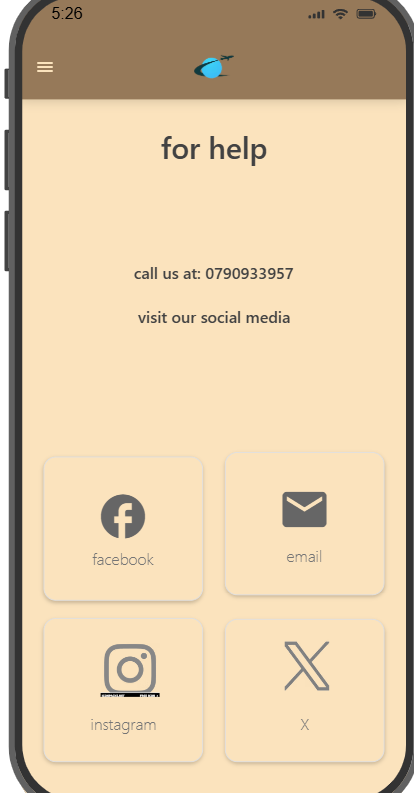
Description automatically generated

A screenshot of a phone

Description automatically generatedA screen shot of a sign

Description automatically generatedA screenshot of a phone

Description automatically generated



# Part 4: Software Quality and Testing

### Discuss two approaches to improving software quality.

Improving software quality is essential to ensure that the system meets user expectations, performs reliably, and is free from defects.

Here are two approaches for improving software quality:

**Early Testing and Defect Detection**

Testing ought to commence early in the Software Development Life Cycle (SDLC), preferably during the design stage. This method seeks to identify flaws early on, stopping them from escalating into bigger, more expensive problems later during the development stage. Initial testing encompasses tasks like requirement analysis, design evaluations, and unit testing. Detecting flaws early greatly lowers the expenses associated with correcting them. For instance, a flaw that requires 100 to repair in the requirements stage could cost 10,000 or higher if identified during the implementation phase. Initial testing also enhances the overall quality of the software, resulting in a more dependable and stable system. Within a tourism booking system, testers can examine the requirements document in the requirement gathering stage to detect possible defects or ambiguities. For example, when the need for the payment gateway integration is ambiguous, testers can identify this problem promptly, guaranteeing that the development team comprehends exactly what must be developed.

**Quality Assurance (QA) and Continuous Monitoring**

Quality Assurance (QA) encompasses the implementation of processes to oversee and confirm that software development actions align with established quality criteria and specifications. QA must be involved throughout the complete development process, from the beginning phases to the final handover. Ongoing monitoring guarantees that quality controls are reliably upheld, featuring routine code reviews, automated testing, and compliance with coding standards. This method aids in recognizing and addressing risks promptly, decreasing the chances of significant problems occurring in the later phases of development. For instance, in the coding stage, developers may create unit tests to ensure that each element of the system (such as user registration and search features) functions correctly. During the maintenance phase, quality assurance processes can be utilized to oversee the system for any problems reported by users, like defects in the payment processing module, guaranteeing prompt resolution and upholding system reliability.

### Evaluate the process of undertaking a systems investigation that you use in Part 2.1 with regard to its effectiveness in improving software quality.

The process of investigating the systems is a vital phase in grasping the needs and features required for the tourism booking software system. In Section 2.1, I performed a systems analysis employing two methods: surveys and background research. In the following section, I will assess how well this process enhances software quality.

**Identifying Key Requirements**

* The systems analysis aided in pinpointing both functional and non-functional requirements for the travel reservation software. For instance, via the survey, I collected user preferences regarding features such as user registration, search and booking, and payment processing. The background research on platforms such as Expedia pointed out industry benchmarks and deficiencies, including the necessity for improved customer service and clearer pricing. By recognizing these requirements early, the development team could concentrate on creating features that satisfy user needs, guaranteeing a user-friendly and effective system. This directly enhances user satisfaction and the reliability of the software.

**Early Detection of Issues**

* The survey uncovered frequent frustrations users encounter with current booking platforms, including hidden fees and slow refunds. The background study also pointed out shortcomings in rival platforms, including inadequate customer support and a lack of transparency. By tackling these challenges early in the design and development stages, the team could introduce solutions such as improved customer support and clear pricing during the reservation process. This minimizes the chances of comparable problems occurring in the end product, enhancing software quality and user confidence.

**Stakeholder Involvement**

* The survey included direct input from prospective users and businesses in the tourism sector. This guaranteed that the collected requirements were thorough and matched both user and business needs. For instance, businesses stressed the significance of compatibility with their current systems and immediate availability updates, whereas users prioritized usability and various payment methods. Engaging a varied group of stakeholders from the beginning guarantees that the software addresses the needs of all involved, minimizing the chances of misaligned requirements and the need for rework later in the development cycle. This results in a superior product that meets the needs of both users and companies.

**Evidence-Based Decision Making**

* The systems investigation delivered evidence-backed insights via survey findings and competitor evaluation. For instance, the survey results indicated that users favor various payment methods (such as credit cards, PayPal, and mobile wallets), whereas the background research uncovered that reward programs serve as a significant differentiator in the marketplace. Leveraging evidence to inform decision-making guarantees that the software is developed according to actual user requirements and leading industry standards. This results in a stronger and more competitive product.

**Areas for Improvement**

* Although the systems investigation proved effective, there are aspects that could be enhanced. For instance, the survey might have included a broader and more varied range of stakeholders, including hotel managers, airline officials, and travel agency representatives. This would have offered a more comprehensive perspective on the needs. Moreover, the background research mainly centered on Expedia, but a more comprehensive evaluation of various competitors (such as Booking.com, Airbnb) might have offered richer insights into industry trends and best practices. These enhancements could additionally boost the thoroughness and standard of the software.

**In conclusion,** **The systems investigation approach utilized in Part 2.1 proved to be very effective in enhancing software quality by pinpointing essential requirements, uncovering problems early, engaging stakeholders, and facilitating evidence-driven decision-making. Nonetheless, there is potential for enhancement regarding wider stakeholder engagement and a more thorough analysis of competitors. By focusing on these areas, the process may further improve the quality and competitiveness of the tourism booking software system.**

### Create a Requirements Traceability Matrix (RTM) document for the tourism booking software (The RTM must be relevant to your implementation in Part 3.4 minimum five functional requirements with two test cases for each one) .

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Req ID | Req desc | TC ID | TC Desc | Test Data | Expected results |
| Req01 | Login and registration | TC01 | Create new using new email and password | [Email=oal@gmail.com](mailto:Email=oal@gmail.com)  Password=123 | Login successful |
| TC02 | Create new using used email and password | [Email=oal@gmail.com](mailto:Email=oal@gmail.com)  Password=123 | Login fail |
| TC03 | Login with correct email and password | [Email=oal@gmail.com](mailto:Email=oal@gmail.com)  Password=123 | Login successful |
| TC04 | Login with invalid email and password | [Email=oal222@gmail.com](mailto:Email=oal222@gmail.com)  Password=123 | Login fail |
| Rq02 | Search | TC05 | Show specific search results based on content | Asia | Show the destinations available there |
| Req03 | Search | TC06 | Display correct information when choosing destination | Bangkok | Shows the right info |
| Req04 | Trip managing | TC07 | Add trip with wanted information | Name  Description  Date  Payment | Trip created successfully |
| Req05 | Trip managing | TC08 | Trip cancelling | Cancel the trip from trips | Trip cancelled successfully |
| Req06 | Destination preference | TC09 | Show the areas based on type | Old city type | Showed successfully |
| Req07 | Destination preference | TC10 | Show user based on his trips | New city | Failed |
| Req08 | Point system | TC11 | Add trip to check for point added | Add trip | Passed |
| Req09 | Points system | TC12 | As admin give reward for user with points | Add 50%off | Successful display in profile |
| Req10 | Profile edit | TC13 | Change profile picture | Change picture | Changed successful |

# Part 5: Evaluation and Justification

### Assess the merits of applying the Waterfall lifecycle model to a large software development project.

The Waterfall model is a fixed, linear methodology in software development, where each phase has to be finished before shifting to the next one. It is mainly used for larger projects that need structure and order, hence its popularity.

Now let’s look into its advantages:

* **Clear Structure and Milestones:** The Waterfall framework has a simple, sequential structure that is beneficial for large undertakings with specific objectives. It breaks the project into smaller phases: requirements, design, implementation, testing, deployment, and maintenance. Each of these phases include assigned objectives and milestones for efficient resource and progress management.
* **Documentation-Driven:** Large projects necessitate extensive documentation for compliance, communication with stakeholders, or maintenance. Due to the structured nature of the Waterfall model, comprehensive documentation is collected at each phase, leading to all the project details being stored at one place for easy access.
* **Predictability:** The Waterfall model necessitates all requirements to be collected at the beginning of the software development life cycle, making the timelines, costs, and deliverables reliable. This happens to be crucial for large projects, where missing the budget and schedule deadline can lead to serious financial loss.
* **Ease of Management**: The sequential nature of the Waterfall Model makes it easy to supervise considerable divisions and very complex projects. Managers are able to allocate resources much more effectively, and the employees can focus on their tasks more because they won’t always be required to alter or change work.
* **Risk Management:** In a big project, there are risks that must be acted upon before they escalate. The Waterfall model enables drafting of blueprints and designs as well as assessing risk in the beginning stages, something that minimizes the chances of other problems occurring during development.

But the Waterfall model does not come without its fair share of weaknesses, particularly for large projects where requirements are even expected to change over the course of development. A downside to this model is its inflexibility, making it difficult to incorporate adjustments that may cause delays and escalate costs if the latter stages have already been entered.

Practical example: The Waterfall model’s strict system has been beneficial in developing systems such as the NASA Space Shuttle software. Users of the models have demonstrated positive responses toward stringent systems accompanied with requirements and complete documentation for balancing the reliability and safety of the software. It was a necessity especially for the life threating project.

### Evaluate how criteria like cost, time, resources, and legal requirements affect the decision-making process and the overall feasibility of the tourism booking software project.

When developing a tourism booking software system, several criteria must be considered to ensure the project's feasibility and success. These criteria include **cost**, **time**, **resources**, and **legal requirements**.

And here is a evaluation how these factors affect decision making.

* Cost
  + Positive side: Effective cost management guarantees that the project remains within budget, facilitating the distribution of resources to essential areas such as development, testing, and marketing. In the feasibility study, the initial expenses for development and marketing were projected to be $25,000, whereas ongoing costs for maintenance and hosting were anticipated at $9,000 each year. These expenses were meticulously assessed to guarantee financial sustainability.
  + Negative side: If costs are underestimated, the project may face financial constraints, leading to delays or compromises in quality. For example, if the actual development costs exceed the estimated **$20,000**, the project may need to cut corners in other areas, such as reducing the scope of features or delaying the launch.
* time
  + positive side: A well-defined timeline ensures that the project is completed on schedule, allowing for timely market entry. The feasibility report outlines a **total project timeline of 186 days**, with specific milestones for each phase of development. This structured timeline helps in managing resources effectively and ensures that the project is completed before the peak travel season.
  + Negative side: Holds in progress can result in lost chances, such as forfeiting market share to rivals. For instance, if the development team does not meet deadlines for essential features such as user registration or payment processing, the launch could be postponed, leading to lost revenue and diminished user trust.
* Resources
  + Positive side: Proper distribution of resources, including qualified developers, designers, and testers, guarantees that the project advances without issues. The feasibility report emphasizes the necessity of possessing a technical team with current skills to guarantee the software's security and functionality. This guarantees that the team can manage the intricacies of the project, such as GDPR compliance.
  + Negative side: Insufficient resources, such as a lack of experienced developers or inadequate hardware, can lead to delays, increased costs, and a lower-quality product. For instance, if the team lacks expertise in AI-powered customer support bots, the implementation of this feature may be delayed or poorly executed, affecting user satisfaction.
* Legal requirements
  + Positive side: Compliance with legal requirements, such as GDPR for data protection and PCI DSS for payment processing, ensures that the software is secure and trustworthy. The feasibility report emphasizes the importance of adhering to these regulations to protect user data and ensure lawful processing. This enhances user confidence and reduces the risk of legal issues.
  + Negative side: Noncompliance with legal obligations may lead to penalties, lawsuits, and harm to the organization's reputation. For instance, if the software does not adhere to GDPR, it may incur fines of as much as 4% of worldwide yearly income, which could negatively impact the project's financial viability.

In conclusion the decision-making for the tourism booking software initiative must thoughtfully consider cost, time, resources, and legal obligations. The feasibility report establishes a strong basis for this equilibrium, featuring comprehensive cost projections, an organized timeline, and an emphasis on legal adherence. Nonetheless, it is essential to observe these elements during the project to guarantee that the software is completed on schedule, within financial limits, and in accordance with all legal requirements. For instance, if the project encounters unanticipated holdups, the team might have to redistribute resources or modify the schedule to prevent undermining the quality of the end product.

### Justify the use of data analytics, machine learning, and other data-driven approaches to enhance the performance, reliability, and user satisfaction of the tourism booking software.

In order to improve the functionality, dependability, and user happiness of travel booking software, data analytics, machine learning, and other data-driven methodologies are essential. Here is a thorough explanation of why they should be used:

* **Personalization and User Experience Enhancement**
  + **For any tourism booking software to function properly, it needs data analytics and machine learning to assist in providing a user-specific experience. And based on these behavioral insights, unique designed recommendations can be served. For instance, if a user books a beach resort too often, the system can feature beaches and related scenery first. Furthermore, different machine learning algorithms can optimize these recommendations by anticipating users’ needs from past data. This makes sense as users are more likely to book vacations when limits are set on how much searching they have to do. Machine learning can also improve the interface as intelligence guides the software, allowing users to reach their goals faster**
* **Improved Decision-Making and Operational Efficiency**
  + **In respect to tourism booking systems, а Data Driven approach increases efficiency and effectiveness. An analyst within a data driven Decision Support System can aggregate information from trends in bookings, customers' market feedback, and other conditions and suggest appropriate strategies. The system can detect the most active booking periods and recommend the implementation of flexible price changes to increase income. Moreover, machine learning models are even capable of estimating future bookings with the help of seasonal statistics and other factors,** **this reduces overbooking or underbooking problems, ensures that the appropriate number of rooms or flights are available at the appropriate time, optimizes inventory management, and boosts overall operational efficiency.**
* **Enhanced Security and Fraud Detection**
  + **A huge amount of money is lost in the tourism industry globally, due to the lack of adequate security on booking platforms. However, data science solutions can improve this considerably. For instance, certain machine learning algorithms can analyze patterns such as sudden shifts in spending, and mark such activities as suspicious. What if a tourist spent lots of money in booking several trips all at once? This can be done by looking for flags set off by users within specific time durations. This type of fraud detection improves service delivery to the clients as well as to the company. Furthermore, data analysis can greatly assist in ensuring compliance with the regulations for example, in the case of GDPR by tracing how data is accessed and used. This, in turn, lowers the threat of data leakage and guarantees that users' data is managed properly, resulting in proper relationships.**
* **Optimized Customer Support**
  + **AI technologies can aid in providing better customer service by improving tourism booking software. Powered by machine learning, AI chatbots can give instant assistance with booking changes and cancellation policies. These services lessen the contact that users need to have with employees and thus save a lot of time. When a business uses this technique, its consumers tend to feel more satisfied due to the shorter response time. Additionally, customer reviews and feedback on the platform can be analyzed using a technique that falls under data analysis, called sentiment analysis. Such actions enable the platform to identify previously noted complaints and effectively resolve them and allow the users to have a pleasant experience, for example, if multiple users complain about hidden fees, the system can flag this issue for the development team to address in future updates.**
* **Enhanced Marketing and User Engagement**
  + **Data analytics and machine learning can also improve marketing strategies and user engagement. By segmenting users based on their booking behavior and preferences, the platform can create highly targeted marketing campaigns. For example, users who frequently book luxury hotels can be targeted with promotions for high-end travel packages. Machine learning can further optimize marketing strategies by running A/B tests on different versions of the platform, such as different layouts or promotional offers, to determine which version performs better in terms of user engagement and conversion rates. This data-driven approach ensures that marketing efforts are effective and aligned with user preferences.**

To sum up, machine learning, data analytics, and other data-driven strategies are critical to enhancing trip booking software. By identifying fraud, they facilitate improved decision-making, individualized user experiences, and increased security. Targeted marketing and AI-powered customer service also increase user engagement and satisfaction. These solutions not only address present issues but also set up the framework for expansion in the future. Travel booking systems may provide a smooth, safe, and easy-to-use experience by incorporating data-driven solutions, guaranteeing company success in a cutthroat industry.

# References

Lomas, N. (2024). *Europe eyes LinkedIn’s use of data for ads in another DSA ask | TechCrunch*. [online] TechCrunch. Available at: https://techcrunch.com/2024/03/14/linkedin-dsa-rfi/ [Accessed 25 Dec. 2024].

Investopedia. (2024). *Search Results from investopedia.com*. [online] Available at: <https://www.investopedia.com/search?q=facebook> [Accessed 25 Dec. 2024].

LinkedIn (2020). *Privacy Policy | LinkedIn*. [online] Linkedin.com. Available at: <https://www.linkedin.com/legal/privacy-policy>.

‌ LinkedIn Talent Solutions (2018). *Recruiting Solutions for the Perfect Hire*. [online] Linkedin.com. Available at: <https://business.linkedin.com/talent-solutions>.

‌ Facebook (2019). *Facebook Business: Marketing on Facebook*. [online] Facebook Business. Available at: <https://www.facebook.com/business>.

‌ Lyskoit, V. (2024). *Facebook data breaches: Investigating top data leaks | NordVPN*. [online] NordVPN. Available at: <https://nordvpn.com/blog/facebook-data-breach/>.

‌ Statista (2023). *Facebook - Statistics & Facts*. [online] Statista. Available at: <https://www.statista.com/topics/751/facebook/#topicOverview>.

‌ Expedia (2024). *Expedia Travel: Search Hotels, Cheap Flights, Car Rentals & Vacations*. [online] Expedia.com. Available at: <https://www.expedia.com/>.

Trustpilot. (n.d.). *Expedia is rated ‘Bad’ with 1.2 / 5 on Trustpilot*. [online] Available at: <https://www.trustpilot.com/review/www.expedia.com>.‌

GeeksforGeeks (2018). *Waterfall Model - Software Engineering*. [online] GeeksforGeeks. Available at: <https://www.geeksforgeeks.org/waterfall-model/>.

‌ GeeksforGeeks (2018). *Software Engineering | Prototyping Model - GeeksforGeeks*. [online] GeeksforGeeks. Available at: <https://www.geeksforgeeks.org/software-engineering-prototyping-model/>.

‌ Kumar Pal, S. (2018). *Software Engineering | Spiral Model*. [online] GeeksforGeeks. Available at: <https://www.geeksforgeeks.org/software-engineering-spiral-model/>.

Kumar Pal, S. (2018). *Software Engineering | Agile Development Models - GeeksforGeeks*. [online] GeeksforGeeks. Available at: <https://www.geeksforgeeks.org/software-engineering-agile-development-models/>.

‌ Taller, H. (2020). *Risk Management in Agile and Waterfall Environments*. [online] www.ptc.com. Available at: <https://www.ptc.com/en/blogs/alm/sdlc-risk-management-in-agile-and-waterfall-environments>.

‌ Moore, K. and Gupta, D. (2011). *Finite State Machines | Brilliant Math & Science Wiki*. [online] Brilliant.org. Available at: https://brilliant.org/wiki/finite-state-machines/.

‌Aggarwal, N. (2017). *An introduction to Flowcharts - GeeksforGeeks*. [online] GeeksforGeeks. Available at: https://www.geeksforgeeks.org/an-introduction-to-flowcharts/.

‌ Kalaji, A.S., Hierons, R.M. and Swift, S. (2011). An integrated search-based approach for automatic testing from extended finite state machine (EFSM) models. *Information and Software Technology*, 53(12), pp.1297–1318. doi:https://doi.org/10.1016/j.infsof.2011.06.004.

Wilcox, K. (2018). *Spotlight on Lessons Learned: Aligning System Development Models with Insight Approaches | APPEL Knowledge Services*. [online] Nasa.gov. Available at: https://appel.nasa.gov/2018/11/27/spotlight-on-lessons-learned-aligning-system-development-models-with-insight-approaches/.

‌

Sandhu, A. (2022). *Advantage Of The Waterfall Methodology [2022]» BestOutcome*. [online] bestoutcome.com. Available at: https://bestoutcome.com/knowledge-centre/waterfall-methodology-advantage/.

‌

Lutkevich, B. (2022). *What is waterfall model? - Definition from WhatIs.com*. [online] SearchSoftwareQuality. Available at: https://www.techtarget.com/searchsoftwarequality/definition/waterfall-model.

‌

Gurnov, A. (2021). *What Is Waterfall Project Management?* [online] Wrike. Available at: https://www.wrike.com/project-management-guide/faq/what-is-waterfall-project-management/.

‌

www.linkedin.com. (n.d.). *How does waterfall development improve risk management in product development?* [online] Available at: https://www.linkedin.com/advice/0/how-does-waterfall-development-improve-risk-erf7f.

‌European Commission (2023). *What if my company/organisation fails to comply with the data protection rules?* [online] European Commission. Available at: https://commission.europa.eu/law/law-topic/data-protection/rules-business-and-organisations/enforcement-and-sanctions/sanctions/what-if-my-companyorganisation-fails-comply-data-protection-rules\_en.

‌

‌

‌

‌

‌