A black background with a black square

Description automatically generated with medium confidence

A black and blue text

Description automatically generated

Student ID: 22210016

Student Name: Raneem Yahya Sadeh

Instructor Name: Dr. Razan/ Eng. Yazan

­

Course Name: Software Development Lifecycle

# Table of Contents

[Table of Contents 1](#_Toc188672359)

[Part 1: Software Lifecycle Models 2](#_Toc188672360)

[1.1 Describe two iterative and two sequential software lifecycle models. 2](#_Toc188672361)

[1.2 Explain risk Management: Definition, Steps, and Risk Strategies in Agile and Waterfall Models 4](#_Toc188672362)

[1.3 Explain the purpose of a feasibility report in software development and discuss the components of a feasibility report. 7](#_Toc188672363)

[1.4 Describe criteria and methods for comparing two technical solutions. 8](#_Toc188672364)

[Part 2: Software Investigation and Analysis 9](#_Toc188672365)

[2.1 Conduct a software investigation to understand the requirements for the tourism booking software by gathering requirements using two techniques (with evidence), analysing current systems, and identifying key functionalities needed. 9](#_Toc188672366)

[2.2 Create a feasibility report for the tourism booking software system. 13](#_Toc188672367)

[2.3 Create a Software Requirements Specification (SRS) document for the tourism booking software system. 16](#_Toc188672368)

[Discuss why you might select a particular lifecycle model for developing a tourism booking software system. 20](#_Toc188672369)

[Part 3: Software Design Techniques and Behavioral Tools 22](#_Toc188672370)

[Discuss using two examples the suitability of software behavioural design techniques. 22](#_Toc188672371)

[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? 23](#_Toc188672372)

[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. 24](#_Toc188672373)

[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. 27](#_Toc188672374)

[Part 4: Software Quality and Testing 35](#_Toc188672375)

[Discuss two approaches to improving software quality. 35](#_Toc188672376)

[Evaluate the process of undertaking a systems investigation 36](#_Toc188672377)

[Create a Requirements Traceability Matrix (RTM) document 37](#_Toc188672378)

[Part 5: Evaluation and Justification 39](#_Toc188672379)

[Assess the merits of applying the Waterfall lifecycle model to a large project. 39](#_Toc188672380)

[Evaluate how cost, time, resources, and legal requirements affect decision-making. 40](#_Toc188672381)

[Justify the use of data-driven approaches like data analytics and machine learning. 40](#_Toc188672382)

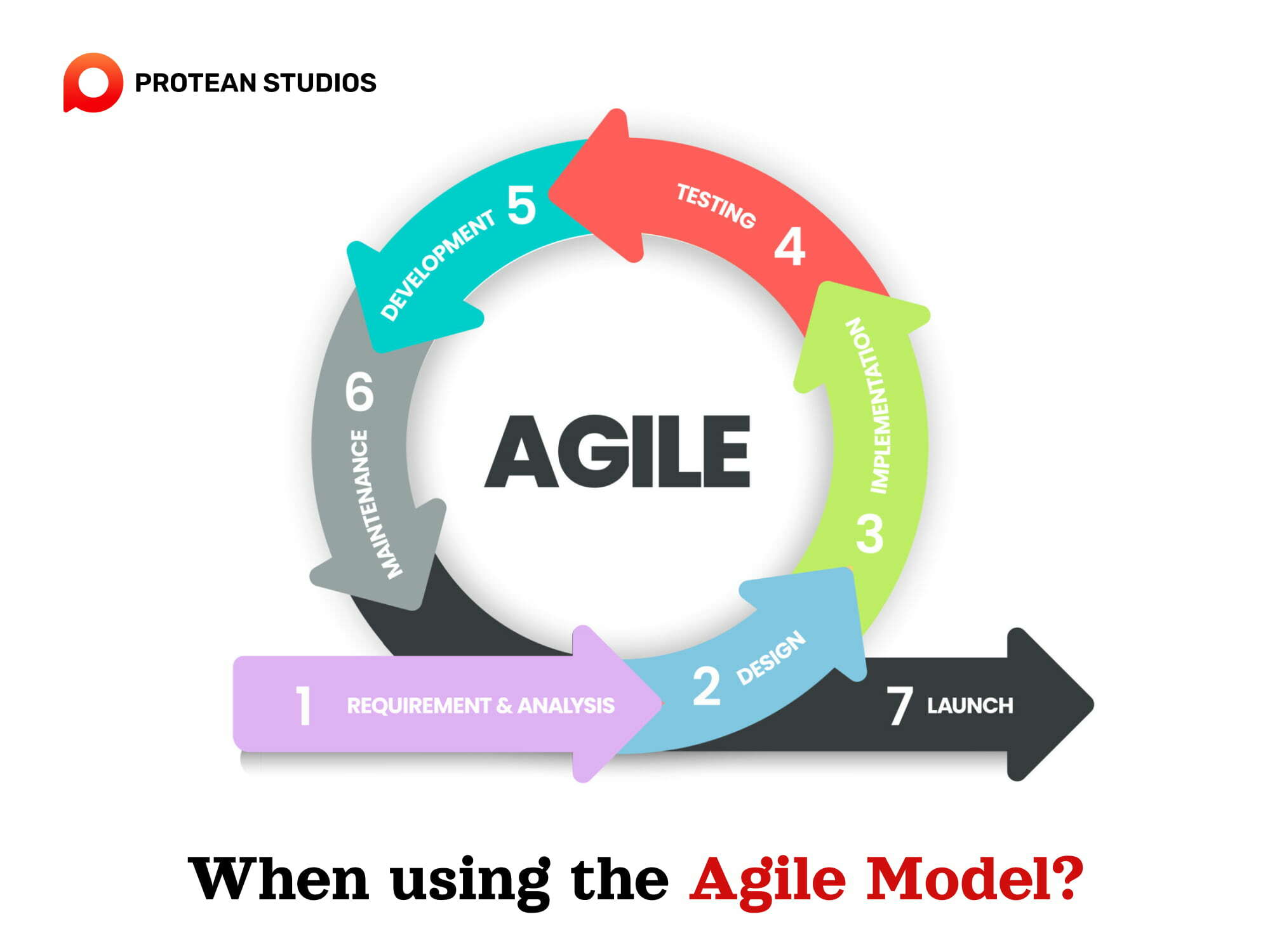
[References 41](#_Toc188672383)

# Part 1: Software Lifecycle Models

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

**Iterative Models**

1. **Agile Model**



If the project is going to be edited or change during its implementation, the best model to use in the project for that is absolutely the Agile Model the goal of this model is to facilitate quick project completion effectuate this task, agility is needed. Agility is attained by making the process to each project, removing any unnecessary steps. and more over, it will avoid anything wasteful of time and effort, The Agile model includes a collection of development processes. While sharing basic traits this processes possess subtle distinctions.

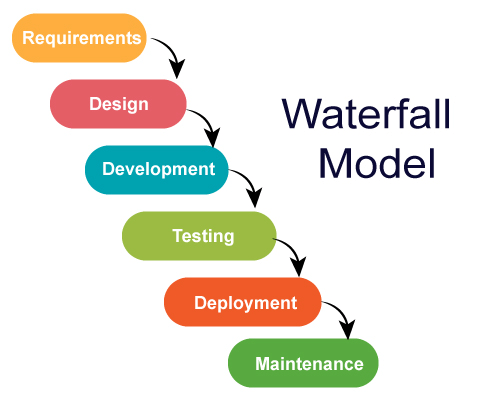
The Agile model is a dynamic and iterative approach to software development that prioritizes adaptability and collaboration and customer feedback. The most important principle of Agile is to deliver functional components of the software in small, incremental cycles known as sprints, each typically lasting 2 to 4 weeks. At the end of every sprint a working module of the software is presented to the stakeholders and allowing for continuous feedback and course corrections about the sprint. Agile encourages teamwork and close interaction between developers and testers and clients to make sure that the product aligns with evolving requirements. Common frameworks within Agile include Scrum, which structures development around roles like Scrum Master and Product Owner, and Kanban, which focuses on visual workflow management. Agile is particularly suitable for projects where requirements are expected to change or where quick delivery of functional features is necessary. For instance, Agile is widely used in mobile app development, where user needs and market demands frequently evolve.

1. **Spiral Model**

**A diagram with green circles

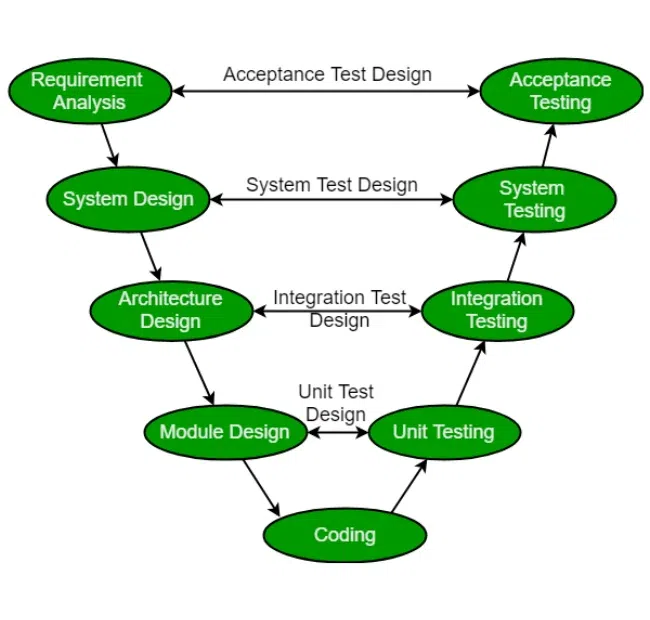
Description automatically generated**  
The Spiral model is an risk driven iterative and repeated development methodology that combines the best aspects of prototyping and the waterfall model. It is structured into a series of spirals and each one of these spiral represents a development phase. These phases include the planning and the risk analysis and the engineering and the evaluation. At each iteration the risks are going to be analyzed and strategies are developed to mitigate them before proceeding to the next phase. This model is going to allow developers to build prototypes that help identify potential challenges early on, and making sure of a robust and refined final product. The Spiral model is particularly well-suited for large-scale and high-risk projects, such as defense systems or mission-critical enterprise software, where the cost of errors is high. Its flexibility enables adjustments at each phase while maintaining a structured approach to development. A key advantage of the Spiral model is its emphasis on stakeholder involvement, as users can review prototypes and provide feedback throughout the process.

**Sequential Models**

1. **Waterfall Model**  
   ****

The Waterfall model is a traditional and sequential approach to software development where each phase including the requirements gathering and the design and the implementation and the testing and the deployment and the maintenance is completed in a linear as one straight line progression and as well each phase serves as a foundation for the next, and once a phase is finalized, there is little room for revisiting or modifying it. This model is best suited for projects with clearly defined and stable requirements, such as government systems or financial software, where changes during development are minimal or undesirable. The Waterfall models structured approach provides clarity and well-documented processes, making it easy for teams to track progress. However, its rigidity can be a drawback in projects where requirements may evolve and more over the testing is conducted only after the implementation phase and as well this can lead to discovering significant issues late in the development cycle.

1. **V-Model (Validation and Verification Model)**

  
The Validation and Verification Model or as known as the V-Model, extends the Waterfall model by integrating testing activities into every phase of development. The model is visualized as a V shape with development stages that are located on the left side and corresponding testing stages on the right as shown in the image. For example lets say that the system requirements are paired with acceptance testing and architectural design with system testing and module design with unit testing. This approach ensures that each phases deliverables meet the required quality standards before moving forward. The V-Model is particularly valuable in safety-critical industries lets say such as the healthcare and aviation, where rigorous testing is essential to make sure of the reliability and compliance. By addressing defects early and consistently throughout the development process, the V-Model reduces the likelihood of costly errors at later stages. Despite its strengths, the V-Models' strict structure may not be suitable for projects with evolving requirements, as it does not readily accommodate changes once a phase is complete.

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

Risk management is the systematic process of identifying and analyzing and prioritizing and mitigating risks that could impact the projects success. While risks’ are often viewed as negative, they can also be positive opportunities lets say such as the chance to expand into new fields or markets. For example lets say that a company venturing into a new service area may face risks related to resource allocation and training and operational adjustments, but this also presents an opportunity for growth and increased revenue and market sharing. Effective risk management make sure perfectly that proactive measures are taken to address threats while maximizing potential benefits and maintaining a balance between minimizing negative impacts and increasing the positive risks to achieve strategic goals.

We have a lot of risks types in software development lets say such as the most famous which are the technical risks and project risks and they can be as Schedule Risk.

**Steps in Risk Management:**

**Risk Identification:** Recognize and knowing potential risks that doe includes the technical and the financial and the operational and the environmental risks.

**Risk Assessment:** Evaluate the likelihood of occurrence and potential impact of each of those risks.

**Risk Prioritization:** Rank risks based on their severity and likelihood to focus on the most critical ones.

**Risk Avoidance and Mitigation Planning:** This is by developing a strategies to prevent or reduce the impact of identified risks.

**Risk Transfer:** This when the company cant solve the risk and handle it by it self so its going to transform this risk to another serves company or provider to solve it, lets say for example there is a security risk and the company provide software app about health care I because the data is really important they need a very good company to solve this security risk.

**Risk Acceptance:** Is when the company decide to accept risk instead of removing or avoiding or mitigating it and this does means that the risk is within the risk tolerance level of this company.

**Monitoring and Control:** and this is by continuously tracking risks throughout the project lifecycle and update mitigation plans as necessary.

For the waterfall and the agile models the steps might be a little different,

**Risk Identification**:

**For the Agile** the risks are identified dynamically and iteratively throughout the project in each sprint or iteration.

**For the Waterfall** the risks are primarily identified during the planning phase and reassessed at key milestones.

**Risk Assessment**:

**In the Agile** the risks are assessed continuously with evolving project scope, priorities, and stakeholder feedback.

**In the Waterfall** the risks are assessed at the beginning of the project with limited opportunities for re-evaluation later.

**Risk Prioritization**:

**In the Agile** the risks are prioritized and incorporated into sprint backlogs as tasks or stories for immediate attention.

**In the Waterfall** the risks are prioritized upfront, and plans are created for critical risks during project planning.

**Risk Avoidance and Mitigation Planning**:

**In the Agil the** Mitigation strategies are developed iteratively, allowing adjustments during each sprint.

**In the Waterfall** the Comprehensive mitigation plans are created at the start of the project, assuming fixed requirements.

**Risk Transfer**:

The process is similar for both models. But also the Agile projects may address transferred risks earlier since they focus on shorter development cycles.

**Risk Acceptance**:

In Agile, risks are accepted iteratively and may be re-evaluated during sprints.

In Waterfall, risks are accepted during the planning phase and rarely revisited unless a major milestone review demands it.

**Monitoring and Control**:

**In the Agile** the Risks are monitored continuously as the team adjusts to changes during sprints.

**In the Waterfall** the Risks are monitored at predefined milestones or through formal reviews.

**Strategies in Agile Model:**

Agile incorporates risk management as an iterative and continuous process. Risks are addressed dynamically in each sprint and by making sure that issues are resolved as they arise and before they escalate.

**Risk Strategies in Agile:**

Incremental Delivery: This is by delivering software in small functional increments that minimize the impact of risks by making sure of continuous progress and early detection of issues.

Frequent Testing: the testing is conducted regularly during each sprint to identify and resolve technical or functional risks early.

Collaborative Communication: the close collaboration between team members and stakeholders and end-users makes sure that potential risks related to scope or requirements are identified and mitigated promptly.

Prioritization via Backlogs: the risks are incorporated into the backlog as tasks or stories making sure that they are prioritized and addressed within the sprint.

Flexibility and Adaptability: the agile flexibility allows for rapid adjustments to project plans in response to risks such as changing requirements or market trends.

**Strategies in Waterfall Model:**

In the Waterfall model, risk management is typically puted during the planning phase. The sequential nature of Waterfall means that risks identified later in the lifecycle can lead to significant project delays or increases of the costs.

**Risk Strategies in Waterfall:**

**Detailed Planning:** a comprehensive planning during the initial stages helps identify and address risks before project execution begins.

**Milestone Reviews:** and by conducting milestone reviews makes sure that risks are monitored at predefined stages, enabling corrective actions to be implemented.

**Risk Mitigation Plans:** for this the detailed mitigation plans are developed for each identified risk, specifying steps to reduce the likelihood or impact of the risk.

**Change Control Processes:** a formal change control process ensures that scope changes are evaluated and their associated risks are mitigated effectively.

**Testing and Validation:** Rigorous testing at the end of the implementation phase is going to help in identify potential risks related to system functionality or integration.

|  |  |  |
| --- | --- | --- |
| Aspect | Agile | Waterfall |
| Approach | This model is iterative and continuous | Its predictive and phase-specific |
| Risk Adaptability | Its high and so on the risks addressed dynamically | Its low and so on the risks is addressed early |
| Planning | Minimal upfront planning; ongoing | Extensive upfront planning |
| Focus on Testing | Ongoing more and more and frequent | Its an End phase testing |
| Collaboration | Its high and the stakeholder and involvement | Moderate its primarily during review |
| Flexibility | For agile its High and Plans adjusted frequently | Its low and changes managed via control |

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

The feasibility report serves as a very important tool in assessing the viability of a software development project by systematically analyzing multiple dimensions such as lets say the technical and economic and operational and legal and scheduling factors.

The goal of the report is to provide a decision makers with a comprehensive evaluation, that help in making them able to determine whether the project is worth pursuing. It acts as a roadmap to identify potential challenges and allocate resources effectively and making sure of the alignment with organizational goals. Beyond viability, a well crafted feasibility report can highlight strategic advantages and suggest optimizations and propose alternative solutions where necessary.

**Technical Feasibility:**

This examines whether the organization has the necessary technological capabilities and expertise to execute the project. This also contains to evaluating hardware and the software and the infrastructure and the personnel. For example lets say that integrating APIs for the tourism booking system would require compatibility checks with existing frameworks and resources for customization and maintenance.

**Economic Feasibility:**

Cost analysis is central to this component, covering the total project expenditure versus anticipated benefits. This includes fixed and variable costs, ROI estimations, and break-even analysis. For example lets say that the deciding between developing in house or outsourcing a tourism app would involve comparing long term maintenance costs and licensing and scalability.

**Operational Feasibility:**

This evaluates whether the system will effectively meet user expectations and integrate smoothly into current workflows. It contains usability testing and employee feedback and stakeholder consultations. Lets say for example when making sure that the apps interface is intuitive for users from different demographics is essential for its adoption.

**Legal Feasibility:**

Assessing compliance with relevant laws and the standards and the policies is vital to avoid legal repercussions, and actually it is one of the most important types. For the tourism booking system this might include GDPR compliance for data security and local tourism regulations and transactional tax laws.

**Scheduling Feasibility:**

Analyzing timelines to make sure that the project can be completed in the specified deadlines that have been puted for example lets say using the gantt chart is is really important for maintaining competitiveness. This involves creating realistic project schedules, including development, testing, and deployment phases. For example lets say that delays in rolling out a booking app before a peak holiday season could result in missed opportunities.

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

**Criteria for Comparison:**

1. **Performance:** Measures the speed, reliability, and responsiveness of the software under varying conditions. for example lets say in a tourism booking software system, performance is critical during peak booking times such as lets say holiday seasons or promotional campaigns. our system must handle a high volume of simultaneous transactions, including user searches, payment processing, and reservation confirmations, without delays or crashes. Poor performance during such periods can lead to customer dissatisfaction, loss of revenue, and potential reputational damage, emphasizing the need for robust infrastructure and efficient algorithms to ensure optimal operation under high demand. Performance metrics may include response time, transaction throughput, and system availability, all of which are essential for ensuring a seamless user experience.
2. **Scalability:** Assesses the ability of a system to handle increased workloads without performance degradation. Scalability is very important for businesses that anticipate growth or face fluctuating user demands. Lets say for example during peak tourism seasons, the booking system must efficiently accommodate a surge in users without delays or failures. Horizontal scalability adding more servers and vertical scalability upgrading existing hardware are common approaches. Lets say like when we evaluating how well each solution accommodates additional users during holiday seasons, including stress testing to simulate high traffic loads.
3. **Cost:** Considers total costs, including development, maintenance, and operational expenses. Cost analysis should account for upfront development fees, ongoing support and maintenance, licensing costs, and potential savings from automation or cloud based solutions. Lets say like when analyzing whether Solution A or Solution B is more cost effective over five years, factoring in hidden costs such as downtime or inefficiencies. Additionally, cost effectiveness can be compared through ROI calculations which determine the financial benefits relative to the investment.

**Methods for Comparison:**

1. **SWOT Analysis:** Identifies strengths, weaknesses, opportunities, and threats for each solution. This method provides a comprehensive view of both internal capabilities and external risks. Lets say like when highlighting Solution A’s advanced features and strong vendor support as strengths but noting its higher costs as a potential weakness. SWOT analysis also helps to identify external opportunities and lets say like market expansion and threats like regulatory compliance challenges.
2. **Benchmark Testing:** Measures performance using standardized tests to evaluate speed, reliability, and efficiency. Benchmark testing involves running predefined scenarios under controlled conditions to determine how each solution performs relative to industry standards or competitors lets say like for example when we are comparing transaction processing times for both solutions during simulated peak load conditions such as lets say a holiday booking rushes to assess reliability and responsiveness.
3. **Cost-Benefit Analysis:** Compares the costs and benefits of each option to determine overall value. CBA provides a quantitative framework for evaluating financial trade offs, ensuring decision makers choose the most economically viable solution lets say like when calculating ROI for Solution A and Solution B based on expected revenues from increased bookings, factoring in savings from automated processes and reduced support costs. CBA can also account for intangible benefits, such as improved customer satisfaction or brand reputation, providing a holistic view of each solution’s value proposition.

# Part 2: Software Investigation and Analysis

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

1. **Scenarios Method**

The scenarios method was utilized to gather insights by developing hypothetical user journeys. These scenarios simulated real life interactions with the tourism booking software, helping to identify key functionalities and potential user challenges. This technique provided a narrative based approach to explore various use cases and ensured that the software aligns with user expectations.

The Sadeh's family, consisting of six members parents Yahya and Alia, and their four children, Hala (12 years old), Rahaf (8 years old), Zakaria (22 years old) and Raneem (20 years old)—planning their first family holiday using the tourism booking app. They wanted a hassle free experience and are all excited to create a customized travel package tailored to their interests in activities, food, and budget.

Hala opens the app and signed up as a new user by providing her email and setting a secure password, ensuring it meets the requirement of being at least six characters long. After she well logging in, she statred with the Home page, which offers two main options and they are the predesigned packages for inspiration for the customers or the users of the app and the ability to build a custom package from zero. And more also, users can explore individual categories such as "Places," or "Food," or "Activities" and book them as one individual options without committing to a full package.

The Sadeh family was really excited to start, the family decides to create their custom package. They begin by selecting their destination. The app displays a many of destinations with a very perfect and high quality images and detailed descriptions. As they scroll, they discuss their shared love for the sea and decide to visit Aqaba which is a popular seaside destination in Jordan. Once they select Aqaba, they move on to the next category and it is the food.

In the food section, Hala eagerly go over through the available options, which feature iconic dishes from Aqaba. And because its Aqaba Sea Food Dish is a must try. The family adds a reservation for a dinner at a well rated seafood restaurant by the beach, known for its grilled fish platters and seafood mezzes.

Next, they explore the "Activities" section as well as he like adventure Zakaria suggests scuba diving since he knows it’s one of Raneem favorite hobbies. They book a family friendly scuba diving session at one of Aqaba’s vibrant coral reef sites, ensuring an unforgettable underwater adventure. Rahaf excited but hesitant about diving, finds a snorkeling option perfect for her. The family also adds a glass in the bottom of the boat tour to their itinerary so everyone can enjoy the stunning marine life.

Finally, they select the Hotel section. They choose a beachfront resort offering spacious family suites and amenities such as a kid's play area, a diving center, and complimentary breakfast. The app allows them to communicate directly with the hotel to arrange connecting rooms for the children.

As they finish the whole package, the app provides a detailed summary, breaking down costs, activities, food, and hotel. It calculates the total cost of the package. They can also put the duration of their stay and the total number of people. After reviewing everything, Yahya the father uses PayPal to pay for the package.

Two days later, Hala log in back into the app to check if they can add another activity to their itinerary. She updates her email in the account settings and saves the changes, appreciating the apps user friendly design. After exploring the activities again, she books a guided cultural tour of Aqabas historical sites.

The findings

User-Centric Features --> the apps flexibility in offering predesigned packages and a custom packages those does enhances user satisfaction and caters to diverse user preferences and the detailed descriptions, high-quality visuals, and intuitive categorization Places," "Food," "Activities" provide a seamless user experience.

Personalized Experience --> the Sadeh family interaction highlights the apps ability to create customized packages tailored to interests and budgets and group needs, such as lets say the connecting hotel rooms or diverse activity options.

Ease of Navigation --> the apps user friendly interface enabled effortless navigation for Hala a first time user from signing up to modifying the itinerary post booking.

Key Functionalities Identified:

Category Specific Options --> the ability to book activities or food or accommodations individually without committing to a full package.

Cost Transparency --> detailed breakdowns of costs ensure clarity for users.

Payment Convenience --> offering PayPal as a payment method enhances accessibility.

1. **Interview Method**

Interviews were conducted with a AI and Software engineer at Tourstify, a leading tourism company that offers a tourism app for visitors to Jordan. The purpose of the interview was to gather insights and information to the systems strengths and limitations, identify key functionalities needed for my tourism booking software, and analyze industry best practices.

A screenshot of a computer program

Description automatically generated

A black text on a white background

Description automatically generated

**The Interview Questions and what Zakria have answered:**

**What features in your app are most popular among users?**

**Zakaria Answer:** The most popular features include personalized recommendations that we gave to the users and the realtime booking for accommodations and activities, and an amazing interactive map showing nearby attractions and services.

**What challenges do users face when navigating your app?**

**Zakaria Answer:** Users sometimes find it challenging to manage their trips across multiple categories, such as accommodation, activities, and transportation. Simplifying the user interface for better flow is a key area we are working on.

**How do you ensure real-time updates for availability and pricing?**

**Zakaria Answer:** We use APIs connected to service providers, ensuring that data on availability and pricing is updated in real time. However, maintaining these integrations can be resource intensive.

**What payment methods do you offer, and how do you ensure secure transactions?**

**Zakaria Answer:** We offer credit cards, PayPal, and mobile payment options. And as well Secure transactions are ensured using encryption protocols like SSL/TLS and PCI compliance standards.

**How do you handle multilingual support for international users?**

**Zakaria Answer:** Our app supports multiple languages, including Arabic, English, and French. Language settings are based on user preferences, and we collaborate with professional translators to ensure accuracy.

**What recommendations would you give for developing a new tourism booking software?**

**Zakaria Answer:** Focus on user experience by providing an intuitive interface and integrating features like personalized itineraries and AI powered suggestions. Also, ensure robust backend support for scalability and security.

**What key features should be included in a tourism app for visitors to Jordan?**

**Zakaria Answer:** Features should include detailed guides to cultural and historical sites, recommendations for local cuisine, real time reservation management, and options for transportation between cities and attractions.

**What is the most requested feature that isn’t yet available in your app?**

**Zakaria Answer:** Users frequently request offline functionality for itineraries and maps, especially for remote areas with limited internet connectivity.

**Key Insights Gathered from the Interview:**

Integration with APIs for real time updates is essential for seamless booking and pricing.

Multilingual support is critical for catering to an international audience.

Simplifying the user interface enhances user engagement and satisfaction.

Adding offline functionality could provide a competitive advantage.

The findings

Popular Features --> personalized recommendations, real-time booking for accommodations and activities, and an interactive map emerged as critical features valued by users.

Challenges Identified --> users face difficulties managing itineraries across categories, underlining the need for a streamlined interface with improved flow between activities, accommodation, and transportation.

Technical Insights --> real time Updates: Integration with APIs ensures accurate pricing and availability but requires resource-intensive maintenance.

Secure Transaction --> encryption protocols SSL or the TLS and PCI compliance are critical for maintaining user trust.

Multilingual Support --> professional translation services and user-configurable language preferences are key to catering to diverse users.

Recommendations for New Software --> focus on intuitive UI/UX design and personalized itineraries and AI-powered suggestions.

Provide offline functionality for itineraries and maps to address connectivity issues in remote areas would be the best.

1. **Requirements gathered from both methods**

**User Registration:**

Secure account creation via email, or phone, providing flexibility and convenience to users.

Integration with user profile management so we allow the users of the app for updating personal details and preferences seamlessly.

**Package Creation:**

Dynamic customization of travel packages where we make the users able to select destinations which is theplaces they want to visit and the food and activities tailored to their preferences and the hotel.

Real-time pricing updates based on user selections.

Availability checks integrated with service providers’ systems to provide users with accurate and up to date information.

Options to save favorite packages for future reference or rebooking where they can check it any time they want.

**Reservation Management:**

Automated booking confirmations sent to users via email and providing instant assurance.

Personalized reminders and alerts for upcoming reservations to enhance user experience and reduce no-shows.

Flexible cancellation options with refund policies clearly communicated to users.

Integration with multiple payment gateways to support a variety of payment methods, including credit cards and digital wallets and bank transfers and making sure of user convenience.

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

This feasibility report explores the practicality and viability of developing and implementing a tourism booking software tailored for visitors to Jordan. With the increasing demand for streemlined and personalized travel experiences and as well this software aim to bridge the gap between user needs and the availability of seamless and user friendly booking platforms. The report investigates the technical, operational, and market aspects of the propose solution, ensuring it aligns with user expectations and industry standerds. Key features such as real time booking and the personalize recommendations and the offline functionality and multilanguage support will be consider to enhance user satisfation and competitiveness. To asses the feasibility comprehensively, this report employs the Scenarios Method and Interviews Methode to gathered insights from potential users and industry experts. The findings highlights the key functionalities and the technical requirements and challenges associated with developing such a solution. This report will serves as a foundation for decision making, evaluating whether the proposed tourism booking software is viable in term of functionality and cost effectiveness and alignment with market demands, ensuring it potential for successful implementation.

***Legal Feasibility:***

Its realy important as its one of the most important types of feasibility because every country has its own laws so if I want to make a project in a specific country it must assign to the laws of that country, for example, my project ‘Tourism system that I am working on MarhabaJordan” its in Jordan so one of the tourism laws on Jordan is that all travel agencies and tourism related businesses must obtain a license from the Ministry of Tourism and Antiquities. This law make sure that the services provided meet national standards and that tourism operations align with cultural and environmental regulations. And more, any data collection or usage in the project must comply with Jordan's data privacy and protection laws, making sure the privacy of both domestic and international users.

**Data protection compliance,** We should make sure that the MarhabaJordan adheres to GDPR for our user data.

**Tourism regulations,** Because the app is only for traveling in Jordan so it must verify compliance with Jordan tourism laws.

***Economic Feasibility:***

In this feasibility we should know all the financial benefits for the app and the cost for developing MarhabaJordan.

**Determining Project Benefits**

First of all we are going to measure the tangible benefits, lets say such as, reducing the personnel expenses and this can be done by automating customer support and bookings by our system, MarhabaJordan can reduce the need for extensive customer service teams, and as well as MarhabaJordans efficient booking system minimizes manual processing costs.

The Improved booking efficiency and user engagement are expected to lead to increased revenue.

The intangible benefits cannot be easily measured in monetary terms but have significant value, lets say like, improving user satisfaction; a seamless and personalized travel planning experience enhances user loyalty. And as well it enhances the national tourism reputation by promoting Jordan's tourism through innovative technology can boost the countrys global image.

And for the Environmental Benefits that Reduced paper use for itineraries and the tickets contributes to sustainability.

To **determine the project costs** using the Tangible Costs where these are measurable and include such as the development costs where it includes the expenses for hiring developers, and designers, and project managers to build MarhabaJordan.

And the hardware costs that purchase servers or cloud services for hosting the application.

Operational Costs: Training employees, maintaining the app, and periodic updates.

For the One Time Cost, costs incurred at the project’s start up phase, lets say such as the systems development and the new hardware and software purchases and the user training and site preparation and the data or system conversion.

For the recurring Costs 🡪 Ongoing costs for maintaining and evolving the app, lets say such as the application software maintenance and the incremental data storage expenses and the incremental communications costs and the supplies and other expenses like additional licenses or personnel.

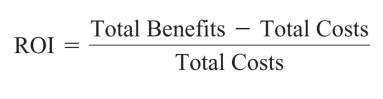
We have three types of **Cash Flow Analysis and the measures** also it makes sure that MarhabaJordan has the financial capability to sustain operations and expand

The first type is the Cash Flows From Operations which tracks the income from app services and subtracts operating expenses.

While the Cash Flows From Investing where it accounts for investment related activities like purchasing servers or upgrading infrastructure.

And at the end the Cash Flows From Financing that works by tracking the funds from investors or capital raised to support the project.

The **Return on Investment analysis** measures the rate of return on the investment in MarhabaJordan.



Here is the equation of the ROI = (Total Benefits - Total Costs) / Total Costs

For MarhabaJordan the Total Benefits: JD 50,000 (from increased bookings and user engagement in the first year)

Total Costs: JD 20,000 (development, hosting, maintenance, and marketing expenses)

ROI = (50,000 - 20,000) / 20,000

ROI = 30,000 / 20,000

ROI = 1.5 or 150%

This ROI indicates that for every JD 1 invested, the project generates JD 1.50 in net benefits. This high ROI demonstrates the economic viability of MarhabaJordan.

***Technical Feasibility:***

A technical feasibility study is going to evaluate the current resources of MarhabaJordan, including the hardware, and the software, and the technical requirements needed for the system. It is going to make sure that the necessary infrastructure is in its perfect place, such as robust cloud services and APIs that are for seamless integration.

The study also examines whether the development team has the required technical skills, lets say such as they are good in Python, SQL, and AWS, as well as other necessary technologies to enhance the application.

And more and more, the study is going to make sure that the technology chosen is easy to maintain and upgrade, and assesses the viability of using current technologies for development. And of course considering the general guidelines for technical feasibility which is going to include the larger projects, is riskier; hence, breaking the project into manageable phases is essential.

Clearly defined and structured requirements will reduce risks compared to vague or ill-structured requirements.

Employing standard and commonly used technologies will lower risks compared to using novel or experimental technologies.

The familiarity of the user group with the system development process and application area will also contribute to reducing project risks.

***Operations Feasibility:***

User Adoption: The MarhabaJordan app must feature an intuitive and userfriendly interface that seamlessly fits into user's daily routines. The goal of all of this is to make sure of the broad acceptance, not just by tourists but also by locals and service providers. Making sure about the ease of use for various demographics will be very important for maintaining high engagement. The app should also be adaptable, making the users able to find relevant information quickly and easily. An essential factor for adoption will be testing with diverse focus groups to ensure usability aligns with realworld needs.

Integration with Existing Tourism Platforms: MarhabaJordan should integrate smoothly with local tourism infrastructure which is going to include the transportation and the accommodations and event booking systems. To achieve this and make it real the app must support seamless data exchange with existing platforms such as lets say the hotel booking services and the local event calendars and the transportation providers. and more over the apps backend should be designed for flexibility, allowing for continuous updates and easy integration with new platforms as they become available. Effective integration will reduce friction for users and service providers, leading to a more streamlined tourism experience.

Operational Impact on Existing Processes: To implement the MarhabaJordan app effectively, operational processes within tourism organizations must be considered. This is going to includes setting up systems for handling bookings and managing real-time recommendations and tracking user behavior. Key departments such as lets say the customer support must be trained to use the app for troubleshooting and assisting users. Local service teams should also be involved in maintaining the apps data accuracy, such as lets say up to date activity schedules and availability and prices. more and more technical support teams will be required to handle system maintenance and updates, ensuring the app's continued reliability.

Scalability Across Multiple Regions: Given the apps target to serve both of the locals and international visitors the scalability will be vital the app should be able to adapt to different countries and cities and considering language barriers and regional preferences and local regulations. This includes multilanguage support and localized content and adherence to local laws such as lets say the data privacy regulations. Each region may require slightly different features or interfaces so the app should be perfectly designed with flexibility for scaling operations.

Ongoing Maintenance and Support: Operational feasibility also involves setting up a robust support system. The apps long term success depends on its ability to adapt to changes whether due to shifts in tourism trends or updates to external platforms it integrates with. A dedicated maintenance team must monitor performance metrics and the address post launch issues promptly and roll out regular updates. This team will also handle customer feedback making sure that any operational glitches are swiftly addressed.

Cost and Resource Management: Operationally, launching and maintaining MarhabaJordan requires significant investment in resources. This includes staff for ongoing app support and as well as physical resources like servers for hosting and infrastructure for data storage and each of these local market may require its own set of operational resources such as lets say the warehouses for physical goods and the local marketing teams and the dedicated customer service reps for in-region support. The overall feasibility hinges on balancing these costs while ensuring that the platform remains sustainable.

Training and Onboarding for Stakeholders: Service providers such as lets say the tour guides and the hotel managers and local vendors will need to be trained to use the app for listing their services on it interacting with users of the app and managing bookings and by making sure that the operational feasibility means making sure that these stakeholders can effectively navigate the platform and are aligned with the app's overall goals. Regular onboarding sessions and technical training and user manuals should be provided to ensure smooth operation.

***Scheduling Feasibility:***

**Timeline:** Estimated completion within six months, including development and testing phases.

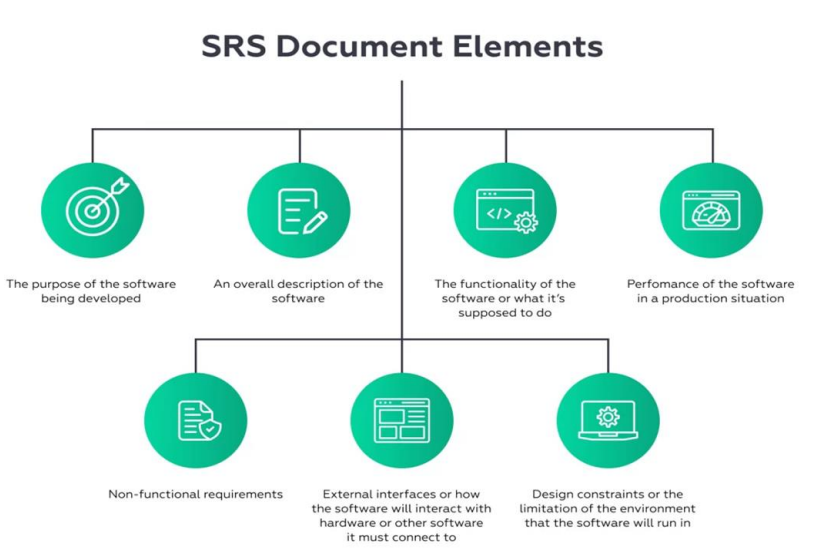
**Milestones:** Weekly sprint reviews to track progress, the analysis indicates that the project is viable, with manageable risks and significant benefits, provided proper planning and resource allocation.

A screen shot of a computer

Description automatically generated

The **Gantt chart MarhabaJordan** project outlines the sequence and timing of tasks necessary for its successful completion. The project is organized into seven key phases which are the Requirement Gathering and Analysis and Resource Allocation and Planning and Technical Feasibility Assessment and Development of MVP and Operational Testing and Adjustments and Implementation Planning and the Launch and Monitoring. Each phase includes specific subtasks starting with gathering essential user and stakeholder feedback and then analyzing current market solutions and at last in this task documenting both functional and non-functional requirements. This information is then used to create a feasibility report. at this sequence first the focus shifts to planning the necessary resources and then including identifying tools and technologies and assembling the development team and lastly assigning roles and responsibilities. With the team in place, the next phase involves evaluating the technical feasibility of the project, including assessing integrations with payment APIs and AI features and verifying infrastructure scalability. Once these technical aspects are sorted the development of the MVP begins, which includes creating basic modules such as user authentication and itinerary creation and integrating essential features like multilingual support and secure payments, and performing internal testing to ensure functionality. After the MVP is ready operational testing follows, where the system undergoes usability testing and feedback is collected from focus groups and app features are refined based on this input. In parallel, the implementation phase includes preparing user guides, training staff, and finalizing a marketing strategy. Finally, the app is launched for pilot testing, followed by monitoring its performance and the addressing any issues and resolving post launch challenges. Throughout this process, several milestones are established including the completion of requirement gathering, the finalization of resource plans, and the successful completion of the proof of concept, MVP readiness, operational testing, and, ultimately, the app launch. The Gantt chart serves as a detailed roadmap for tracking the projects progress and making sure that dependencies are respected and the resources are efficiently utilized the and all necessary tasks are completed within the specified timeline.

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



**Description**

**Purpose**

The purpose of this Software Requirements Specification (SRS) document is to outline the functional and non-functional requirements of the MarhabaJordan Tourism Booking System. This system is designed to provide users with a seamless experience for planning and booking their trips within Jordan, with features tailored to enhance user convenience, promote eco-tourism, and support local businesses.

**Scope**

MarhabaJordan is a tourism booking platform that caters to both domestic and international travelers. The system offers comprehensive features, including customizable travel packages, real-time reservation management, and secure payment options. The platform’s primary goal is to promote tourism in Jordan by providing:

Real-time availability that the admin how is responsible to check it always of Hotels, activities, and transportation.

Integration with local guides, cultural experiences, and eco-friendly options.

The system will be available as a mobile application, ensuring accessibility for users across various devices (Andriod and IOS).

**Definitions, Acronyms, and Abbreviations**

SRS means 🡪 Software Requirements Specification

API means 🡪 Application Programming Interface

UI/UX means 🡪 User Interface/User Experience

Payment Gateway means 🡪 A service that authorizes and processes online payments

**Overview**

I’m going to provide details for the requirements for the development of MarhabaJordan, where it is going to cover the functional, and the non-functional, and the system-level specifications. All these requirements goal is to guide the development team in delivering a user-centric and efficient solution where they should develop a perfect app including all the requirements as mention.

**Product Perspective**

MarhabaJordan has been made to be look as a centralized platform for tourists to plan, book, and manage their travel experiences in our amazing country Jordan. The system offer a lot of thing and this includes:

The integration with third-party service providers lets say such as the hotels, and in the future will be with the airlines, car rental services.

And as well it should provide a highly secure environment for user data and transactions.

Include a feedback that we get from the comments in the app store and the play store and any other feedback about the app to mechanism to continuously improve features based on user input.

**Product Features**

**For the User Registration:**

The user must could make an account using his email, and username and he should write a password and he might put a profile image and this is optional.

The user could always update his profile by going to the profile sittings and manage it and see his username and email and name from his profile.

**Travel Package Customization:**

The user of the app could create his own customized package for travelling by selection the place he want to visit and the hotel and food and activity he want to make.

Real-time pricing updates where the admin is responsible to check always if the data is updated for the hotels and any serveces the app provide.

**Reservation Management:**

View and modify bookings via a dedicated dashboard for the hotels because the app provide the number of the hotels so the user could contact them directly.

In the future there will be a notifications for updates, weather changes, and promotions.

Integration with multiple payment gateways lets say such as the credit card or other platforms like paypal.

**Local Integration:**

And this servec should be studied for the future where we might make collaboration with local guides, transportation providers, and cultural organizations.

And of course the app support for eco-tourism initiatives and environmentally friendly options.

**User Classes and Characteristics**

Tourists: Primary users who plan and book trips.

Service Providers: Businesses offering accommodations, tours, and transportation.

Administrators: Personnel managing content, resolving issues, and ensuring system performance.

**Operating Environment**

The system will operate on:

Mobile Platforms: iOS and Android.

Server Environment: Cloud-based infrastructure ensuring scalability and reliability.

**Design and Implementation Constraints**

Adherence to data privacy laws lets say such like the GDPR for intenational users.

Compliance with PCI DSS standards for payment security.

Localization support for Arabic and English languages.

**Assumptions and Dependencies**

Reliable internet connectivity is assumed for optimal system performance.

**User requirements:**

-the app should have a interface which is Easy to navigate.

-Secure login and data handling for the user is really important to have a very high security and privacy.

-the users could a customizable packages based on their preferences.

-users can easly edit there informations at the account page in the app.

-getting the best and available offers in the app.

**Functional Requirements**

**User Registration:**

Users must be able to create accounts using an email address, phone number, or social media.

Password recovery through email or SMS.

**Package Customization:**

Users should be able to customize travel packages based on preferences.

Dynamic pricing updates as users add or remove package elements.

**Booking Management:**

Users can view, modify, or cancel bookings.

Notifications for booking confirmations and updates.

**Payment Processing:**

Integration with multiple payment methods (credit/debit cards, digital wallets).

Support for currency conversion for international users.

**Local Features:**

Recommendations for local attractions and activities based on user preferences.

Integration with eco-tourism initiatives.

**Non-Functional Requirements**

**Performance:**

The system must support at least 10,000 concurrent users.

Response time for user actions should not exceed 2 seconds under normal load.

**Security:**

Data encryption for all sensitive information.

Regular vulnerability assessments.

**Usability:**

Intuitive UI/UX design for both web and mobile applications.

Accesibility features for users with disabilities lets say such as the screen readers, high contrast mode.

**Scalability:**

The system must scale horizontally to handle increased user demand during peak seasons.

**Reliability:**

99.9% uptime guaranteed through cloud-based architecture.

**System Requirements**

**Hardware Requirements:**

**Client Devices:**

Smartphones with at least 2GB RAM and modern browsers.

Desktop/laptops with minimum 4GB RAM.

**Server Infrastructure:**

Cloud hosting with auto-scaling capabilities.

**Software Requirements:**

Client Side: React.js for web, Swift/Java for mobile.

Server Side: Node.js backend with Express framework.

Database: PostgreSQL with support for geospatial data.

**APIs:**

Integration with payment gateways, weather services, and airline booking systems.

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

For developing MarhabaJordan, which is a tourism booking software system specifically designed for visitors to Jordan, the **Agile Model** is the most suitable choice due to its iterative and adaptable nature. Agile provides the necessary flexibility to meet the unique demands of tourism in Jordan, where trends such as seasonal activities, new travel regulations, or emerging eco-tourism initiatives can influence user needs. For example, integrating features like real-time availability of eco-friendly accommodations or personalized cultural tours can be achieved swiftly using Agiles dynamic approach.

Agiles sprint-based delivery ensures that functional increments are continuously released, allowing the team to introduce new features like integration with local guides or updated maps of tourist hotspots, such as Petra or Aqaba, without disrupting ongoing operations. This iterative process also enables rapid adaptation to user feedback, ensuring that features like customized travel packages, local cuisine recommendations, and adventure activity bookings align closely with user expectations and market demands.

And more and more, Agile emphasizes collaborative communication, making it easier for developers to work closely with stakeholders, including tourism authorities, local businesses, and end-users. Regular review meetings ensure that all stakeholders remain aligned, while feedback loops help refine functionalities like real-time weather updates for hiking trips or secure payment options for international travelers. By fostering strong collaboration between technical teams and tourism operators, the Agile Model ensures that MarhabaJordan not only meets current user needs but also remains scalable and adaptable to future demands in the evolving tourism industry of Jordan.

Agile Vs Traditional SDLC Models

Agile is an repetitive and adaptive software development methods, whereas the traditional SDLC models like the waterfall model is based on a predictive approach. Predictive teams in the traditional SDLC models usually work with detailed planning and have a complete forecast of the exact tasks and features to be delivered in the next few months or during the product life cycle.

Predictive methods entirely depend on the requirement analysis and planning done in the beginning of cycle. Any changes to be incorporated go through a strict change control management and prioritization.

Agile uses an adaptive approach where there is no detailed planning and there is clarity on future tasks only in respect of what features need to be developed. There is feature driven development and the team adapts to the changing product requirements dynamically. The product is tested very frequently, through the release iterations, minimizing the risk of any major failures in future.

Customer Interaction is the backbone of this Agile methodology, and open communication with minimum documentation are the typical features of Agile development environment. The agile teams work in close collaboration with each other and are most often located in the same geographical location.

Agile Model - Pros and Cons

Agile methods are being very very much accepted and used in the software world recently. However, this method may not always be suitable for all products. Here are some pros and cons of the Agile model.

The advantages inherent in the Agile Model are enumerated below:

It constitutes a highly realistic approach to software development.

Promotes teamwork and cross training.

Functionality can be developed rapidly and demonstrated.

The resources required are few.

Adaptable to both static and dynamic specifications.

Provides preliminary functional solutions in advance of deadlines.

Fit model for environments that change steadily.

Minimal rules, documentation easily employed.

Facilitates parallel development and deployment within a pre-defined framework.

Minimal planning is necessary.

Easy to manage.

Gives flexibility to developers.

The disadvantages of the Agile Model are as follows −

The risk of sustainability comes higher, maintainability and extensibility.

The project will not work if there isnt an agile leader and agile PM practice.

There is a so much high individual dependency, and all that is because there is minimum documentation generated.

# Part 3: Software Design Techniques and Behavioral Tools

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

Two key software behavioral design techniques suitable for the MarhabaJordan tourism booking system are:

In the MarhabaJordan tourism booking system, two key software behavioral design techniques which are the State Machine Modeling and the Sequence Diagrams and ther are particularly suitable.

the SMM

State Machine Modeling is a powerful technique for defining the various states that an application or system can be in and the transitions between those states. It is usaulyy useful in managing the booking lifecycle and visualizing the different steps that are involved. which include the Initial Search where the user enters their travel preferences and searches for available packages and then after reviewing options the user selects a package they are interested in.

at the payment processing the user proceeds to payment and the system processes the payment details.

at the end the booking confirmed after successful payment, the system confirms the booking and provides confirmation to the user.

This technique helps in clearly defining the flow between these stages, particularly when handling transitions like moving from a payment failure state to a retry state, or when successfully transitioning to a booking confirmation state. It also aids in error handling by providing clear definitions of what happens when an invalid operation is attempted.

mine while for the suitable behavioral design diagram the Finite State Machine (FSM) is an extension of State Machine Modeling and fits perfectly in this context. An FSM could defines a system in terms of a finite number of states, transitions between those states, and the actions that occur during transitions. For example lets say in the payment processing state of MarhabaJordan, the FSM would manage transitions between states like:

Idle → Payment Entry 🡪 User inputs payment information.

Payment Entry → Processing Payment 🡪 The system sends the payment details to the gateway.

Processing Payment → Payment Failed or if the Payment Successful 🡪 If the payment fails or succeeds, the system transitions accordingly.

This process continues, ensuring that each state is appropriately handled.

and at the end we ahve the EFSMs extend FSMs by incorporating more detailed conditions such as lets say the variable checks or complex event processing. For example, lets say in our MarhabaJordan it could use EFSMs to manage conditions like does the user's card have sufficient funds? or lets say is the user's location eligible for the selected tour package? These additional inputs and conditions would enhance the FSM's capabilities, making the transition and behavior logic more nuanced.

Sequence Diagrams

Sequence Diagrams are used to model the interactions between system components over time. They illustrate how objects or components communicate with one another to perform a specific function or task. In the case of the MarhabaJordan system, sequence diagrams can model the flow of operations between the user interface, booking engine, and payment gateway, providing clarity on how the components work together to complete the booking process lets say for example the user interaction where they can submits a search query through the user interface and the booking engine interaction where the UI sends the query to the booking engine, which processes it and returns available packages ans as well the payment gateway interaction and after the user selects a package, the UI sends the payment details to the payment gateway for processing dn at the end we have the system response where the payment gateway sends a response (success/failure) back to the UI, which then informs the user of the booking status.

This technique helps identify potential communication bottlenecks and streamline interaction flows. It’s particularly useful for visualizing complex operations like real-time availability checks or booking confirmation. It also could helps developers identify performance issues such as lets say if there are delays in communication between the booking engine and the payment gateway.

Suitable Behavioral Design Diagram --> the DFDs are helpful for representing the flow of information within the system and at the context level, the DFD would depict the major processes involved in MarhabaJordan such as lets say the booking and the payment processing and the user authentication, along with their associated data stores available packages, payment records. At level one, DFDs would break down these processes into smaller more detailed steps showing the interactions between system components in more granular detail lets take this as a example the Context Level DFD shows the overall interaction between the user, the booking system, and the payment gateway on the other hand, the Level One DFD breaks down the booking process into smaller sub-processes, such as user login and package selection and payment details entry and payment processing this is useful for developers to make sure that the data flows correctly through the system and for identifying potential issues or redundancies in the process.

Flowcharts can be used to describe the specific decision-making process in the booking lifecycle, lets take this example if the user inputs invalid data during payment so the flowchart would show that the process loops back to the Payment Entry state.

If the payment is successfully processed the flowchart would transition to the Booking Confirmed state.

### 3.2 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?

**State Variables and Data**

As I have write above and more the FSM does only tracks basic states without any of the additional variables. For example lets say that in the booking, states are simply "Available" or "Booked"

For the EFSM but in the other hand this includes variables and conditions. So as well it could track number of guests and room types and pricing variations within states of it

**Transition Complexity**

FSM have a Simple transitions between states lets say for example from “Searching" to "Booked"

EFSM: Complex transitions with guards and conditions lets say for example transition to "Booked" only if payment successful AND rooms available)

**Examples in Tourism Booking:**

FSM Application for the Room Availability

* States: Available, Reserved, Occupied, Maintenance
* Object: Hotel Room
* Transitions: Simple status changes

EFSM Application in the Booking Process

* States: Searching, Selected, Payment Pending, Confirmed
* Object: Booking
* Variables: Number of guests, dates, days, total cost
* Conditions: Payment verification, availability checks

### 3.3 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.

This document outlines the software design specification for the **MarhabaJordan App**, which will provide a comprehensive platform for users to plan and book travel experiences in Jordan. The system includes functionalities such as package creation, booking management, and payment processing

Which include **User Interface (UI)** where its a front end interface that allows users to search for and book tourism packages and the **Engine** which is a backend system that manages the details of available packages and performs booking management tasks and interfaces with the payment gateway and the **Payment Gateway** which is an external service that securely processes payment transactions.

The **Functional Requirements** for MarhabaJordan focus on providing a amazing user experience. Users must be able to register and log in and edit and manage their profiles and update the account. They can search for travel packages based on preferences and its with details like destination, itinerary, cost, and accommodation provided. Additionally, users will have the ability to add packages to their cart, check real-time availability, and securely process payments through an integrated gateway, generating booking confirmations and receipts. The system features AI-powered tools, such as an itinerary planner, real-time availability check, and personalized recommendations.

The **Non-Functional Requirements** ensure the system performs optimally under high user loads and is secure. It also for sure must support up to 10,000 concurrent users with a real time availability checks and booking confirmations completed in less than two seconds. Scalability is essential to accommodate growing user demands, while robust security measures, including data encryption and SSL protocols, are required to protect user and payment information. The user interface must be responsive across all devices, ensuring simplicity and ease of use. The system must be available 24 hour daily, with daily data backups and an uptime of 99.9%.

For the data bases, we have the user where it have (username, password where its hashed, email) and the package, cart, and the food and the hotels and the places and activities and all of them are very important for the app that the package creation cant be without the food and hotel and places and activities data bases and when the user create the package it will be saved in the packages data base.

For the **UI Wireframes** we have the **Home Page** which includes thenavigation bar with options for search, user login, and package categories and the featured packages items shown with images and pricing and the **Package Details Page** which includes a detailed description of the package, itinerary, reviews, and availability and the bottom to add to cart or book now at the end we have the **Checkout Page** which includes the summary of selected packages, user details, and payment options and the real time availability check.

The Finite State Machine

A diagram of a flowchart

Description automatically generated

The context level of the DFD

|  |  |
| --- | --- |
| A diagram of a diagram  Description automatically generated  The level zero for the DFD | |
|  | |

### 3.4 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.

|  |  |
| --- | --- |
|  |  |
|  | The first page that you will see when opening the app and you can chose if you want a new account or you already have an account to login |
|  | Here is the sign up page where you are creating a new account to use the app |
|  | You have an account? Perfect, this is the page of logining in with your account. |
|  | Don’t forget so save the environment, a message that you will see before getting to the home page |
|  | Finally the home page where you can see a created packages or create you own package |
|  | And as well you can see the categories that you will choose form the package items |
|  | The creating package page, when you press the create bottom in the home page it take you to this page to create you own. |
|  | The food page from the package creation |
|  | The activities page from the package creation, and there is as well the Places page and the hotels page |
|  | The navigation, where you can go to the home page and the settings and log out from you account and down there in you name you can go to the profile page. |
|  | You can see you account information and edit it from the account bottom, and see you package and complete it to checkout from the package bottom |
|  | Here when you choose the items you want you should fill the following which are the package name and the number of days and people to get the final price of you package |
|  |  |
|  |  |
|  | This appears when you press checkout bottom where you fill them and you booking is done |
|  | And alllll done |
|  |  |
|  | Here after finishing editing you press update and you account will be updated |
|  | A message to confirm you decision in logging out |
|  |  |
|  | A message to confirm you decision in updating |

The URL and the code for مرحباJordan, enjoy it 😊

A screenshot of a phone

Description automatically generated

<https://ranemo-sadehs-team.adalo.com/marhabajordan>

# Part 4: Software Quality and Testing

### 4.1 Discuss two approaches to improving software quality.

**Implementing Quality Assurance**

QA processes that does have a systematic activities designed to to make sure that the software meets defined quality standards throughout its lifecycle, such as defining standards by adopting industry standards such as the ISO 9001 or the CMMI to maintain consistent quality, and ofcousrse the test planning to develop detailed test plans for each development phase and including unit testing, integration testing and user acceptance testing and as well to make continuous monitoring and regularly review and audits to identify and address quality issues early.

**Automated Testing:**

it is going to enhances efficiency and accuracy and this by reducing manual errors. and this could be done using tools like Selenium and JUnit and Appium can use to perform regression testing to make sure that a new updates do not introduce defects and aslo test application scalability by simulating multiple users accessing the system simultaneously also as well conduct security tests to verify data protection measures such as encryption.

**Encouraging Innovations:**

Build a culture of innovation is going to allow teams to explore new tools and techniques and methodologies that improve software quality and as well it does encoure brainstorming sessions and hackathons that would lead to creative solutions for complex challenges.

**Communication is Key:**

Effective communication always makes sure that all team members and stakeholders and end-users have a clear understanding of project goals and quality standards as well regular meetings and progress updates and feedback loops are essential for identifying and addressing quality concerns promptly.

**Long-Term Strategy:**

by using and making a long-term perspective that helps maintain quality over the softwares lifecycle and this is going to includes planning for scalability and conducting periodic audits and implementing continuous integration and delivery (CI/CD) pipelines to ensure the system evolves without compromising quality.

### 4.2 Evaluate the process of undertaking a systems investigation

The process of undertaking a systems investigation for the مرحباJordan system involved a structured approach to identify key objectives, evaluate the effectiveness of the system, and makes sure about the stakeholder satisfaction while balancing cost and time efficiency.

**Objectives of the Systems Investigation**

The objective of the systems investigation for the tourism booking software is to understand user needs, identify current system limitations, and define requirements for developing an efficient, user-friendly application by lets say gathering and documenting functional and non-functional requirements and analyzing existing tourism software systems to identify best practices and gaps and making sure of the alignment with user expectations and business goals by using structured investigation techniques.

**Evaluation of the Investigation Techniques**

first we focus in creating a easy to use booking platform as mentioned the UX is very important to focus in and the user interface and the navigation will make the user experience better, also the functionalities in the app must be evaluted to make sure that the system is able to handle complex queries such as the creation of packages, and for sure the performance of the app such as lets say the speed and response time and its working under a huge load. Also its a must to make sure about the smooth integration with external APIs and databases for real-time updates.

**Assess Stakeholder Satisfaction**

The stakeholders, including tourists, tourism service providers, and government authorities, were engaged throughout the development process. and as well their satisfaction was measured through their regular feedback to align the systems features with user expectations and as well when sharing prototypes during development to gather input and refine features, and in testing where we make an implementing the software in a limited capacity to address any concerns before full-scale deployment.

**Measure Software Quality Improvements**

and this is done through the conducting the unit testing and integration testing and user acceptance testing to identify and resolve bugs and as known the best is to use the key performance indicators such as reduced booking errors and faster processing times and higher user retention rates and for the future enhancements we should evaluate the performance of AI features like the chatbot and trip personalization for accuracy and relevance.

**Consider Cost and Time Efficiency**

The systems development was monitored for cost efficiency --> so the budget optimization was achieved by re using existing resources where possible and leveraging open-source AI tools and as well the time management --> where the Agile methodologies have make sure of the timely delivery of project milestones, with a focus on high-priority features in the initial phases.

**Review Documentation and Deliverables**

Comprehensive documentation was maintained to make sure of the clarity and continuity so lets start with the technical documentation where a detailed descriptions of system architecture and APIs and algorithms and also there must be guides for both end-users and administrators to operate the system effectively and for sure the regular status reports to update stakeholders on progress and challenges.

**Learn from Feedback and Lessons Learned**

first the importance of stakeholder involvement where the engaging the stakeholders early helped to align the project with user needs and also that the feedback loops enabled continuous improvement of system features and as well the proactively addressing potential risks such as data security and system scalability improved overall project outcomes.

### 4.3 Create a Requirements Traceability Matrix (RTM) document

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Sno | Requirement ID | Requirement  Description | Test Case ID | TC Desc | Test Design | Test Designer | UAT Test Req | Test Execution | | |
|  | | | | | | | | Test Env | UAT Env | Prod Env |
| 1 | Req01 | Login to the App | TC01 | Sign up without email | Completed | YXZ | YES | Passed | Passed | Passed |
| TC02 | Login with invalid username | Completed | ZXY | YES | Passed | Passed | Passed |
| 2 | Req02 | Create Package in the App | TC03 | User can add places and activities in the package | Not Completed | ZYX | NO | Passed | No run | Passed |
| TC04 | User can add multi food to the package | Completed | ZYX | YES | Passed | Passed | Passed |
| 3 | Req03 | Checkout the package | TC05 | User can add the number he want of days and people | Not completed | YZX | NO | Passed | No run | No run |
| TC06 | The user can see the package final price when finishing the package | Completed | XZY | YES | Passed | Passed | Passed |
| 4 | Req04 | Editing the profile | TC07 | User can update his username from profile | Completed | YXZ | YES | passed | Passed | Passed |
| TC08 | User can update the profile photo | Completed | YXZ | NO | Passed | No run | Passed |
| 5 | Req05 | Login out and delete the account | TC09 | User can delete his account | Completed | ZXY | YES | Passed | Passed | Passed |
| TC10 | User can logout from the account | completed | XYZ | YES | Passed | Passed | Passed |

The Requirements Traceability Matrix is a tool that links requirements to their respective test cases to ensure comprehensive test coverage. The Sno (Serial Number) column provides a unique sequential reference for each entry. The Requirement ID is a unique identifier for each system requirement while here the Requirement Description explains the functionality being tested the Test Case ID column maps requirements to specific test cases, and the TC Desc (Test Case Description) provides a brief overview of the test scenario.

The Test Design column indicates the completion status of test case preparation. The Test Designer column identifies the individual responsible for creating the test case. The UAT Test Req column clarifies whether the test case is required for User Acceptance Testing (UAT). The Test Execution section details results from three environments: Test Env (Test Environment), UAT Env (User Acceptance Testing Environment), and Prod Env (Production Environment). This section shows whether the test case was executed successfully ("Passed"), not executed ("No run"), or failed. The RTM always make sure of traceability from requirements to testing which highlights any incomplete testing and provides a comprehensive overview of test execution status across environments. It supports effective communication between stakeholders and makes sure that all system requirements are met with appropriate testing.

# Part 5: Evaluation and Justification

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

The Waterfall model, with its linear and sequential approach, offers the following advantages for large projects:

* **Clear Structure:** Each phase lets say for example the requirements and design and implementation is well-defined, providing a clear roadmap for the team.
* **Thorough Documentation:** Comprehensive documentation ensures that all stakeholders have a clear understanding of the projects goals and progress.
* **Ease of Management:** Progress is easy to track as one phase must be completed before moving to the next.

**Real-Life Case**

SAP SE, a global leader in enterprise application software, developed its flagship SAP Enterprise Resource Planning software using a Waterfall-like approach. The project was initiated to address the need for an integrated system capable of managing core business functions like finance and procurement and supply chain and human resources.

The linear, phase-by-phase progression ensured clarity and organization. Each phase had specific deliverables, making it easy to track progress and as well this approach was particularly beneficial for SAP ERP, as the software catered to highly regulated industries like banking, healthcare, and manufacturing.

but this is not forever and because of the lack of flexibility, as client needs evolved, SAP found it challenging to incorporate changes without disrupting the project timeline. This limitation was particularly evident in adapting to emerging technologies like cloud computing during later stages and also the testing and client feedback occurred late in the development cycle, leading to delayed identification of usability issues.

Phases of the SAP ERP Development Process

first of all the requirements gathering and planning, stakeholder engagement --> SAP worked closely with enterprise clients to gather comprehensive functional and non-functional requirements, The primary goal have been to design a system capable of managing a business critical processes across industries and as well a clear documentation which is going to detail requirement specifications were created, and making sure of the alignment with diverse business needs like financial reporting standards, international tax laws, and supply chain optimization.

and for testing --> individual modules were tested in isolation to ensure functionality and the integrated system was tested for data flow, user interactions, and process execution as well also SAP conducted rigorous user acceptance testing (UAT) with enterprise clients to identify gaps and ensure usability.

### 5.2 Evaluate how cost, time, resources, and legal requirements affect decision-making.

**For the** **Cost**: Cost analysis is very important in helping prioritize essential features and which makes sure that the development stays within budget. For example lets say that choosing open-source AI tools for the AI chatbot reduces overall expenses while still providing quality functionality on the other hand the budget constraints can create challenges when considering advanced features. If the budget is limited so it may force the team to reduce or eliminate certain advanced functionalities or reduce the scope of testing, which could negatively impact the overall software quality. A careful balance must be struck between cutting costs and maintaining the quality and competitiveness of the app.

**For the Time**: Time is a critical factor in making sure that the app is developed and delivered on schedule. And by setting realistic and achievable timelines this allows for the timely release of key features such as lets say the AI-powered itinerary planner which is a central part of the apps value proposition. However on the other hand the overly rigid deadlines can pose a risk. A compressed timeline may push the team to rush development, potentially compromising important aspects of the app, such as lets say thorough testing, debugging, and quality assurance. If there isnt enough time to adequately test all features, the app’s overall quality could suffer leading to user dissatisfaction or technical issues post-launch.

**For the Resources**: The availability of skilled developers and specialized AI tools and cloud infrastructure significantly accelerates the development process and improves the apps overall capabilities. By having access to experienced team members and the right tools can lead to a more robust and feature rich app. In the other hand when resources are limited, the development process may slow down. The lack of skilled personnel or unavailable tools may result in delays or force the team to scale back on certain features or reducing the apps competitiveness or delaying its market readiness. In such cases resource constraints may also require prioritizing basic functionality over innovative or advanced capabilities.

**For the Legal Requirements**: Legal compliance is vital especially when handling sensitive user data. Adhering to regulations such as lets say the GDPR that does make sure that user privacy and data security are respected and which helps in building trust and credibility with users. On the other hand the process of complying with legal requirements can increase both development time and costs. For example lets say integrating secure data storage systems and implementing privacy safeguards and conducting thorough audits can add significant complexity to the development process. These legal obligations might extend timelines and raise costs but the tradeoff is the ability to operate in global markets with confidence that the app respects data privacy laws and is trustworthy for users.

### 5.3 Justify the use of data-driven approaches like data analytics and machine learning.

Data-driven approaches like data analytics and machine learning significantly enhance the functionality of booking software.

Using data analytics in a software system can greatly improve customer satisfaction and business operations. In order to gain valuable insight into the booking process, it is important to track key metrics lets say such as conversion rates and abandonment rates and average booking values.

Recognize patterns and trends by learning about popular services, customer demographics, and peak booking times.

Predicting resource requirements from booking data can help increase operational efficiency by avoiding overbooking or underutilization.

Find underperforming services and upselling opportunities to improve your offerings and boost revenue.

Apply data to create a personalized experience for customers by customizing services to their preferences.

Here are some examples of how could I enhance MarhabaJordan using machine learning techniques and the data driven by analysing and getting insights,

**Personalized Schedules 🡪** Machine learning algorithms analyze user preferences and historical data to create tailored itineraries. For example the machine learning feature “زهقان” suggests activities based on user moods and interests.

**Matching Solo Travelers 🡪** AI models cluster users with similar travel goals, improving their travel experiences by connecting them with like minded individuals.

**AI Chatbot 🡪** A chatbot offers real-time assistance, addressing user queries and providing recommendations, reducing the need for human customer support.

**Improved Decision-Making🡪** Data analytics identifies patterns in user behavior where it does enable service providers to optimize offerings and marketing strategies, and by predicting using the machine learning models we can take decisions based on facts.

Platforms like Airbnb use data-driven approaches to recommend accommodations and activities based on user preferences. Where its absolutely similar to the tourism booking software leverages AI to enhance user satisfaction and operational efficiency, setting it apart in the competitive travel market.

# References

*Adalo, n.d. Feature Templates. Available at:* [*https://app.adalo.com/apps/f505bc10-0955-499a-bd6d-cd050cdb5338/feature-templates/374900c5-ef93-46fa-9ecf-614b2f2d7792*](https://app.adalo.com/apps/f505bc10-0955-499a-bd6d-cd050cdb5338/feature-templates/374900c5-ef93-46fa-9ecf-614b2f2d7792)*.*

*AltexSoft, n.d. Quality Assurance, Quality Control, and Testing: The Basics of Software Quality Management. Available at:* [*https://www.altexsoft.com/whitepapers/quality-assurance-quality-control-and-testing-the-basics-of-software-quality-management*](https://www.altexsoft.com/whitepapers/quality-assurance-quality-control-and-testing-the-basics-of-software-quality-management)*.*

*Asana, n.d. OKR vs. KPI. Available at:* [*https://asana.com/resources/okr-vs-kpi*](https://asana.com/resources/okr-vs-kpi)*.*

*Colorado.edu, n.d. Software Design. Available at:* [*https://home.cs.colorado.edu/~kena/classes/5828/s10/presentations/softwaredesign.pdf*](https://home.cs.colorado.edu/~kena/classes/5828/s10/presentations/softwaredesign.pdf)*.*

*GeeksforGeeks, n.d. Software Analysis and Design Tools. Available at:* [*https://www.geeksforgeeks.org/software-analysis-and-design-tools/*](https://www.geeksforgeeks.org/software-analysis-and-design-tools/)*.*

*GeeksforGeeks, n.d. Software Engineering | Agile Development Models. Available at:* [*https://www.geeksforgeeks.org/software-engineering-agile-development-models/*](https://www.geeksforgeeks.org/software-engineering-agile-development-models/)*.*

*GeeksforGeeks, n.d. Software Engineering | Software Quality. Available at:* [*https://www.geeksforgeeks.org/software-engineering-software-quality/*](https://www.geeksforgeeks.org/software-engineering-software-quality/)*.*

*GeeksforGeeks, n.d. Types of Feasibility Study in Software Project Development. Available at:* [*https://www.geeksforgeeks.org/types-of-feasibility-study-in-software-project-development/*](https://www.geeksforgeeks.org/types-of-feasibility-study-in-software-project-development/)*.*

*GeeksforGeeks, n.d. Waterfall Model. Available at:* [*https://www.geeksforgeeks.org/waterfall-model/*](https://www.geeksforgeeks.org/waterfall-model/)*.*

*Harvard Business Review, n.d. The Agile Manifesto Explained. Available at:* [*https://hbr.org/2020/03/the-agile-manifesto-explained*](https://hbr.org/2020/03/the-agile-manifesto-explained)*.*

*IEEE Computer Society, n.d. Software Requirements Specifications. Available at:* [*https://www.computer.org/resources/software-requirements-specifications*](https://www.computer.org/resources/software-requirements-specifications)*.*

*Indeed, n.d. Requirement Gathering Techniques. Available at:* [*https://www.indeed.com/career-advice/career-development/requirement-gathering-techniques*](https://www.indeed.com/career-advice/career-development/requirement-gathering-techniques)*.*

*Jama Software, n.d. 11 Requirements Gathering Techniques for Agile Product Teams. Available at:* [*https://www.jamasoftware.com/requirements-management-guide/requirements-gathering-and-management-processes/11-requirements-gathering-techniques-for-agile-product-teams*](https://www.jamasoftware.com/requirements-management-guide/requirements-gathering-and-management-processes/11-requirements-gathering-techniques-for-agile-product-teams)*.*

*Medium, 2021. Why User Personas Matter for Design. Available at:* [*https://uxdesign.cc/why-user-personas-matter-for-design-355bf115631c*](https://uxdesign.cc/why-user-personas-matter-for-design-355bf115631c)*.*

*Perforce, n.d. How to Write a Software Requirements Specification (SRS) Document. Available at:* [*https://www.perforce.com/blog/alm/how-write-software-requirements-specification-srs-document*](https://www.perforce.com/blog/alm/how-write-software-requirements-specification-srs-document)*.*

*TechTarget, n.d. Software Requirements Specification. Available at:* [*https://www.techtarget.com/searchsoftwarequality/definition/software-requirements-specification*](https://www.techtarget.com/searchsoftwarequality/definition/software-requirements-specification)*.*

*TherapyRide, n.d. Understanding the Software Development Life Cycle (SDLC). Available at:* [*https://medium.com/@therapyRide/understanding-the-software-development-life-cycle-sdlc-7e06e37a90a3*](https://medium.com/@therapyRide/understanding-the-software-development-life-cycle-sdlc-7e06e37a90a3)*.*

*TestFort, n.d. QA, QC & Testing: The Basics of Quality Management. Available at:* [*https://testfort.com/blog/qa-qc-testing-the-basics-of-quality-management*](https://testfort.com/blog/qa-qc-testing-the-basics-of-quality-management)*.*

*TutorialsPoint, n.d. SDLC - Waterfall Model. Available at: https://www.tutorialspoint.com/sdlc/sdlc\_waterfall\_model.htm.*

*Tuleap, n.d. Software Quality: Different Types of Software Testing. Available at:* [*https://www.tuleap.org/software-quality-different-types-software-testing*](https://www.tuleap.org/software-quality-different-types-software-testing)*.*

*UX Planet, 2022. How to Conduct Usability Testing for Apps. Available at:* [*https://uxplanet.org/how-to-conduct-usability-testing-for-apps-c8e4e1cba62e*](https://uxplanet.org/how-to-conduct-usability-testing-for-apps-c8e4e1cba62e)*.*

*Atlassian, n.d. Agile Project Management. Available at:* [*https://www.atlassian.com/agile*](https://www.atlassian.com/agile)*.*

*DZone, n.d. Best Practices for Software Quality Assurance. Available at:* [*https://dzone.com/articles/best-practices-software-quality-assurance*](https://dzone.com/articles/best-practices-software-quality-assurance)*.*

*Lucidchart, n.d. Benefits of Using Gantt Charts. Available at:* [*https://www.lucidchart.com/blog/benefits-of-gantt-charts*](https://www.lucidchart.com/blog/benefits-of-gantt-charts)*.*

*PMI, 2020. The Value of Gantt Charts in Project Planning. Available at:* [*https://www.pmi.org/learning/library/gantt-chart-project-planning-11645*](https://www.pmi.org/learning/library/gantt-chart-project-planning-11645)*.*

*Smartsheet, 2023. Gantt Chart Templates and Examples. Available at:* [*https://www.smartsheet.com/gantt-chart-templates*](https://www.smartsheet.com/gantt-chart-templates)*.*

*Wrike, n.d. How to Use Gantt Charts for Agile Teams. Available at:* [*https://www.wrike.com/blog/gantt-charts-agile/*](https://www.wrike.com/blog/gantt-charts-agile/)*.*

*Forrester, 2021. The Impact of AI on Software Development. Available at:* [*https://go.forrester.com/blogs/impact-ai-software-development/*](https://go.forrester.com/blogs/impact-ai-software-development/)*.*

*Google Developers, n.d. Best Practices for App Development. Available at:* [*https://developers.google.com/*](https://developers.google.com/)*.*

*IBM, 2022. Machine Learning Models in Software Applications. Available at:* [*https://www.ibm.com/*](https://www.ibm.com/)*.*