**Submitted to:**

**Dr. Durgansh Sharma**

**Submitted by:**

**RICHA YADAV (86)**

**SOUVIK BARDHAN (59)**

**SHASHANK SHEKHAR (51)**

**BCA (2ND YEAR)**

**Github**-<https://github.com/Shashank02-tech/SAFAR/blob/master/agile%20(1).docx>

Table of Contents

[Pert chart 5](#_Toc37707399)

[Process flow diagram 7](#_Toc37707400)

[ABSTRACT 8](#_Toc37707401)

[INTRODUCTION 9](#_Toc37707402)

[OBJECTIVE 10](#_Toc37707403)

[SUB-OBJECTIVE 10](#_Toc37707404)

[UML DIAGRAM 11](#_Toc37707405)

[11](#_Toc37707406)

[METHODOLOGY 12](#_Toc37707407)

[There are six steps involved in Agile Development Methodology: - 13](#_Toc37707408)

[1. Requirement Analysis 13](#_Toc37707409)

[2. Design 13](#_Toc37707410)

[3. Development 13](#_Toc37707411)

[4. Quality Assurance 13](#_Toc37707412)

[5. Deployment 14](#_Toc37707413)

[6. Release to market 14](#_Toc37707414)

[Law of software evolution 15](#_Toc37707415)

[Law of continuing change 15](#_Toc37707416)

[Law of increasing complexity- 15](#_Toc37707417)

[Law of conservation of organization- 15](#_Toc37707418)

[Law of conservation of familiarity- 15](#_Toc37707419)

[Manifesto for Agile Software Development 15](#_Toc37707420)

[Individuals and interactions over processes and tools 15](#_Toc37707421)

[Working software over comprehensive documentation 16](#_Toc37707422)

[Customer collaboration over contract negotiation 16](#_Toc37707423)

[Responding to change over following a plan 16](#_Toc37707424)

[Principles of Agile Software 16](#_Toc37707425)

[Customer satisfaction through early and continuous delivery of the product. 16](#_Toc37707426)

[Accommodate changing requirements. 17](#_Toc37707427)

[Frequent delivery of working software from a couple of weeks to a couple of months, with a preference to the shorter timescale. 17](#_Toc37707428)

[Business people and developers must work together daily throughout the project. 18](#_Toc37707429)

[Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done. 18](#_Toc37707430)

[Prefer face-to-face communication over other methods 18](#_Toc37707431)

[Working software is the primary measure of progress: 19](#_Toc37707432)

[Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely. 19](#_Toc37707433)

[Continuous attention to technical excellence and good design enhances agility. 19](#_Toc37707434)

[Simplicity-the art of maximizing the amount of work not done is essential. 20](#_Toc37707435)

[The best architectures, requirements, and designs emerge from self-organizing teams. 20](#_Toc37707436)

[At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behaviour accordingly. 20](#_Toc37707437)

[Software characteristics 21](#_Toc37707438)

[Functionality 21](#_Toc37707439)

[Suitability- 21](#_Toc37707440)

[Accurateness 21](#_Toc37707441)

[Interoperability 21](#_Toc37707442)

[Compliance- 21](#_Toc37707443)

[Security- 21](#_Toc37707444)

[Reliability 21](#_Toc37707445)

[Recoverability 22](#_Toc37707446)

[Fault tolerance 22](#_Toc37707447)

[Maturity- 22](#_Toc37707448)

[Efficiency 22](#_Toc37707449)

[Time behaviour- 22](#_Toc37707450)

[Resource behaviour 22](#_Toc37707451)

[Usability 22](#_Toc37707452)

[Learnability 22](#_Toc37707453)

[Operability 22](#_Toc37707454)

[maintainability 23](#_Toc37707455)

[Testability 23](#_Toc37707456)

[Stability 23](#_Toc37707457)

[Changeability 23](#_Toc37707458)

[Operability 23](#_Toc37707459)

[Portability 23](#_Toc37707460)

[Adaptability 23](#_Toc37707461)

[Instability 23](#_Toc37707462)

[Replaceability 23](#_Toc37707463)

[Data flow diagram 24](#_Toc37707464)

[**Level 0** 24](#_Toc37707465)

[Level 1 25](#_Toc37707466)

[Leve 2 26](#_Toc37707467)

[ER diagram 27](#_Toc37707468)

[Class diagram 28](#_Toc37707469)

[Sequence diagram 29](#_Toc37707470)

[29](#_Toc37707471)

[State chart 30](#_Toc37707472)

[Scrum methodology 31](#_Toc37707473)

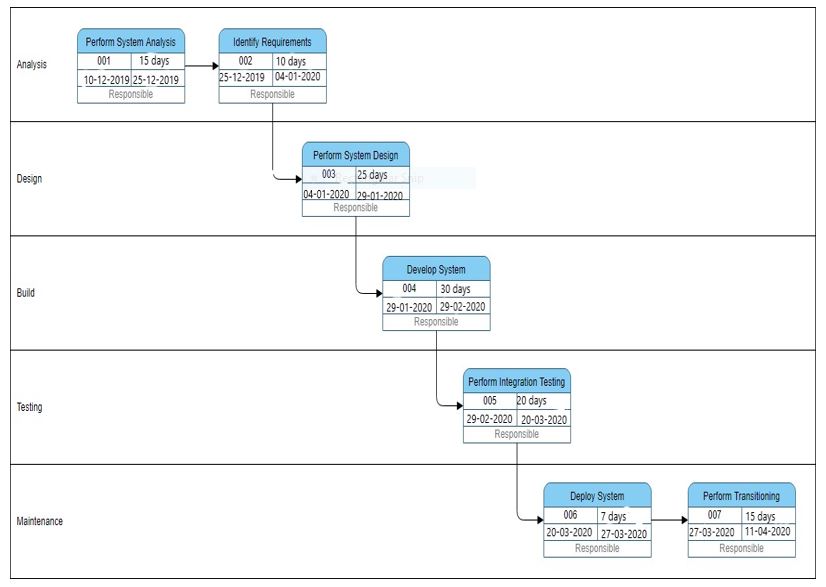
[The Components of Agile Scrum Development. 32](#_Toc37707474)

[The Scrum Team 32](#_Toc37707475)

[**Scrum Events (Ceremonies)** 33](#_Toc37707476)

[Scrum Artifacts 34](#_Toc37707477)

# Pert chart



**People :**

Our application Safar, tend to satisfy the people who are the end users. They are the ones who will ultimately use our application and are going to provide us with valuable feedback. The application that we are going to offer to the end users will satisfy their needs and will help them to execute a variety of tasks in the process that will make our product more efficient and effective at the same time. The product will satisfy the requirements of the people by making their travelling experience hassle free. If our product strikes the interest of the travellers, we will earn goodwill and at the same time we will gain confidence as well. Similarly, a process that is well executed will help in the proper development of the product. The processes are planned in a manner such that it will motivate the users to travel more often and use our application to plan their entire trip. A well planned process helps us to minimize the errors. The end users will save a lot of money and time by using our application and can explore without any worries. People are the heart and soul of our application as the sole purpose is to develop an application that will be used by them to plan their entire journey.

Process

In phase 1 of our project we discussed the Software Development Lifecycle and Agile Development Methodology as they form the foundation for developing our product. Most importantly we discussed Manifesto for Agile development. The 4 points are the key features of agile systems. We prepared data flow diagrams of our project that indicates the flow of data as well as entity relationship model. Now, the phase 2 comprises of the PERT chart (Program Evaluation and Review Technique) and the software processes. The process leads to the development of a satisfactory product and executes certain tasks. Software engineering focuses on the aspect of process. Processes when executed in a planned way yield better results and achieve better project objectives. Software process is used to produce a software that is high in quality and at better costing. Two major processes include:

1. Development: focuses on development of a software.

2. Project Management: focuses on the planning that is made at the time of the development stage.

The term ETVX is used in processes and specify the steps that is well laid out in our product:

1. Entry criteria: It specifies what conditions should we take care of while initiating the phase of our project.

2. Tasks: what is to be done in the phase of our project.

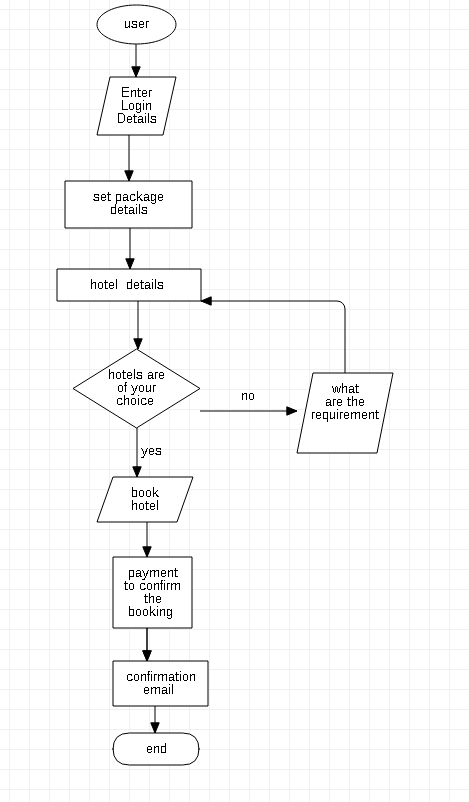
3. Verification: the steps taken and final checks done that weather our product is working fine or not.

4. Exit criteria: when this phase can be considered successful.

Technology:

Our product will use the latest version of the technologies so that it can provide the users with better results. The application should be efficient enough to handle the requests of all the customers. In order to achieve this feat we will acquire a large database to store information from the users. The technology used should be error free and easy to use by end users. Technology gives birth to any application and how efficient the app is it depends upon the reliability of the technology used.

# Process flow diagram



# ABSTRACT

Safar, our application aims to make travelling easy for the wanderlust. In today’s digital world all the bookings and planning are done before-hand to ensure a stress free journey. They have to use various applications to confirm the bookings. Our application combines all the important aspects of travelling into a single application and charge reasonable prices to our fellow travellers.

Our project follows two famous software development methodologies – Scrum and Agile Methodology. It strictly adheres to the agile manifesto in order to provide the customers with a satisfactory end-result.

In order to clearly explain the functioning of our application various diagrams are used such as DFD, ER, State diagram, Sequence diagram and many more. Our project is well incorporated with the agile principles as well as characteristics.

Our team Nemesis has strictly followed the Agile method of development to create a product that is reliable and updated with changing needs. It is made sure that no error occurs and helps as many travellers as possible.

# 

# INTRODUCTION

**“The world is a book and those who do not travel read only a page.” ~ Saint Augustine.** People often travel to escape from the chaos of their daily life. Avid travellers often travel to gain a new perspective about life as well as to find their inner soul. People with family and children often travel during vacations to spend quality time with their family. Some travel out of curiosity while some travel to spend time with their loved ones. But there is common thing among all types of tourists that they travel to give themselves a break from their monotonous schedule and enjoy the vibrant beauty of the world.

**‘Safar’** is our application that will help the tourists to sort and plan their whole trip at one place. Some people are tight on budget so they often postpone their plans to accumulate more money. By using our application, they can know the cost of the trip before-hand and can travel without paying extra money to the cab drivers, hotels etc.

People travel to relax so their experience should be as simple and convenient as possible. They can book a cab, book hotel rooms, know about the history of the places they want to visit with animated interactive videos and many more using just our application. In this way, they can save themselves from the greedy guides and drivers who charge a bomb from the tourists.

The software will be developed using Agile Development Methodology so that we can update with it with ever increasing demand of the user. It will be easy to use and fully tested to so that the users does not face any inconvenience. Through simultaneous deployment and quality assurance we will fulfil all the requirements of the user.

This project consists of Unified Modelling Language (UML), Data Flow Diagram (DFD) and Entity Relationship Diagram (ERD) that explains the structure as well as the functionality of the software. The whole project is incorporated with principles of the Agile so that the end-product will be as realistic and efficient as possible.

# OBJECTIVE

The primary objective of the application is to provide a convenient and budget-friendly travelling experience to the users so that they can travel without any hassle. It’s a step towards revolutionizing the tourism industry.

Guides often charge higher prices from the tourists. To save their hard-earned money they can use the application to eliminate the need of guides. The most important objective of our application is to provide the users with online guide. With interactive video sessions and detailed elaborate explanation of the history, the users will enjoy their exploring experience to the fullest.

# SUB-OBJECTIVE

**The secondary objectives of our application are as follows: -**

To provide the users with low price cab services which they can book before-hand. It will save a lot of time and effort as it eliminates the changing of cabs to go to different places.

To reserve room in the hotels and enjoy their stay to the fullest. The tourists can stay near the tourist spots by booking rooms using our application.

# UML DIAGRAM

# 

# METHODOLOGY

There are various methods in which a software can be developed such as Waterfall Model, Agile Model, Incremental Model etc. Agile Development Methodology will be used to develop our application, **‘Safar’**.

Agile Methodology is a practice that promotes continuous iteration of development and testing throughout the software development lifecycle of the project. Both testing activities are concurrent unlike the Waterfall model.

**The agile software development emphasizes on four core values: -**

Individual and team interactions over processes and tools.

Working software over comprehensive documentation.

Customer collaboration over contract negotiation.

Responding to change over following a plan.



## There are six steps involved in Agile Development Methodology: -

1. Requirement Analysis **-**The first step of the method is to identify the requirements that are expected from the software. Requirement analysis focuses on the tasks that determine the needs or conditions to meet the new or altered software, taking account of the requirements of the investors, analysing, documenting, validating and managing software. Requirement analysis is critical to the success or failure of a systems or software project.

**For example -**We will carefully analyse the needs or requirements of the investors as well as the users of our application to create a satisfactory end-product. The inputs of the users can be taken in order to know what they need from our application by conducting surveys through the beta version of the application. If a majority of the users think they require list of restaurants near the tourist places, we can always make changes to fulfil their needs even after the application is released unlike the Waterfall model.

1. Design **–** The next step involves designing the software. The standard designing process divides the product preparation period into various phases, which, separates interface design and its implementation. The combination of both phases and the creation of the one team working in Agile methodology allows us to save time, speed up the implementation process and improve the quality of a product’s usability. Our application will be designed in such a way that it is appealing to the users. ‘Safar’ will be easy to use as the design of the layout will be simple and organized so that the users can easily enjoy all the features such booking a cab, reserving hotel rooms, knowing the price of the tickets etc. in the most simplest manner.
2. Development – The most important step is the development phase. This process is divided among various teams handling their area of expertise. All the teams work towards achieving the common objective of creating an end-product that meets all the requirements. Each team will perform their assigned tasks such as the technical team will write the codes, the designing team will create a good design to make it user-friendly, the marketing team will find unique ways promote our application etc. The development phase will be executed cautiously and skilfully by our teams so that it may function properly when released for the user. The workplace will also have all the necessary equipment to facilitate a fully functional software. If any changes are to be made in the software after the development phase, we can always update it in Agile methodology.
3. Quality Assurance – Quality Assurance (QA) is a systematic process that ensures product and service excellence. A robust QA team examines the requirements to design, develop, and manufacture reliable products whereby increasing client confidence, company credibility and the ability to thrive in a competitive environment. Our team will begin the QA process at the inception of the software development life cycle. From the initial design meeting, through the development phase, to final testing and hardening of the application. The process will be repeated in two weeks sprints until the project is released. The satisfaction of the customer will be of utmost importance to us, so we will make sure the software is fully tested and in working condition. All the attributes of our application such as booking a cab, reserving a hotel room, checking prices of tickets should be working properly so that it does not cause any inconvenience to the users. We will use the all he features by ourselves before releasing the application for user download.
4. Deployment **–** Agile deployment may be seen as simply another testing step since multiple development deployments are performed between production deployments. QA team are deeply involved in improving the system by providing frequent feedback. We will deploy the software (at initial stage) to an environment that is accessible to specific users and as close as possible to a real-world environment. In this way, users can continuously test the software and send it back for improvement. We can also deploy a beta version to a limited number of users in the final environment for a greater insight into the real-world usage. Continuous deployment and testing of the software will make it more efficient and will also help us to identify the errors in it. The beta version can also be used as a way to know about the general public’s response regarding our application.
5. Release to market– The final step involves releasing the software into the market. A lot planning is done before releasing the product for the use of public. Agile release cycles should be certainly kept shorter than a year, and are often as short as 6 months or 3 months. In order to make the release of our application a success, it will be delivered to users, or at least a subset of users, incrementally at the end of each iteration or every couple of iterations. After listing the initial feature list, our marketing team will hold a release planning meeting to establish an overall release schedule and determine which features can be likely to be delivered. The marketing will also promote our application so that a majority of the public can know about it. Providing discounts and vouchers at the beginning will also compel the users to use our application to sort out their vacation plans.

# Law of software evolution

Law of continuing change – it states that any software system that represents some real world reality undergoes continuous change or become progressively less useful in that environment.

Law of increasing complexity-as an evolving program changes, its structure become more complex unless effective efforts are made to avoid this phenomenon.

Law of conservation of organization- over a lifetime of a program, the rate of development of the program is approximately constant and independent of the resources devoted to system development.

Law of conservation of familiarity-it states that during the active lifetime of the program

, changes made in the successive release are almost constant.

# Manifesto for Agile Software Development

## Individuals and interactions over processes and tools

Individuals and teams put the focus on people and their energy, innovation, and ability to solve problems. if you value processes and tools, the benefits are that the processes are generally clear and well-understood, and you have a written record of communications about the project.

## Working software over comprehensive documentation

Working software means that other than documentation there should also be a small model for a client to get an idea of the type of product he is getting at the end. Comprehensive documentation is not more useful in the terms.

## Customer collaboration over contract negotiation

Customer collaboration is that when the team and the client negotiate when the work is not complete on time then the client comes up with some changes and some more time for the full completion of the project.

## Responding to change over following a plan

The project team responding to the current changes which makes them more relevant and helps in making a useful product which people may use.

# 

# [Principles of Agile Software](https://agilemanifesto.org/principles.html)

## Customer satisfaction through early and continuous delivery of the product.

Customers get to see the product only after completion and when several tests and quality checks have been performed. This not only keeps the customers in dark but also makes it troublesome for the team members to introduce any changes in the product, so in case of our application, **Safar**, we would upload the beta version on application store of different operating systems.

The customers will use our application and provide us with valuable feedback so that we can continuously improve the application on the basis of customer’s requirement. This would make them happy as their inputs have been taken into consideration for the development of the application.

## Accommodate changing requirements.

The development process requires changes to be accommodated at various stage. Our application provides customers with the entire history of all the tourist places in India along with cab and hotel services. In the near future, if the customers does not require any one of our services and want us to replace it with some more useful service then we can accommodate the changing requirements. For example, if the customers does not feel the need of using our cab service anymore and want us to replace it with a list of restaurants near the tourist places, then we have to pay heed to their demands and make the required changes .These changes can help clients gain a competitive edge and also prevent delays.

## Frequent delivery of working software from a couple of weeks to a couple of months, with a preference to the shorter timescale.

This explains how to provide immediate value to the customers by delivering working features. The teams ensure that each feature is fully developed tested,

customised, and styled according to the customer’s satisfaction before considering it as delivered. If the customers want some changes in the application then we will update the app and deliver it to them earlier or in the promised time.

## Business people and developers must work together daily throughout the project.

A major problem associated with traditional project management methodologies is that the stakeholders are often oblivious to the development stages of the project. We will avoid this problem in the case of our project the investors and the developers will have a continuous interaction session with each other so that they can receive feedback on the development of the project and can provide the inputs if they want to. This would result in a good and satisfactory end-product.

## Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.

Fruitful and competitive projects depend on focussed, trusted, and motivated individuals to get the job done. The members associated with our projects will be allowed to select the work they are most interested in by self-organisationwith no interference of external management.

It will be our job to create an encouraging and motivating environment where the members will be not afraid to voice their opinions and give suggestions for the betterment of the team’s performance. Giving the team members incentive and motivating them will cause their overall performance to improve significantly which will ultimately be better for our project.

## Prefer face-to-face communication over other methods

#### This form of interaction is the best one of the lot. No other mode of communication could beat this one, especially when you need to get to the root of an issue. To discuss any problem or change in our project, for effective communication, methods like memos and email are not preferred and more importance will be given to face-to-face communication or video conference as it always encourages a smoother transfer of information amongst the members rather than coming to the office for a short meeting.

## Working software is the primary measure of progress:

#### The only factor to measure success is the delivery of a working product that satisfies the customer. Before Agile, there were many measures of success and that resulted in a drop in the quality of the final product. Our primary concern should be to develop our project in such a manner that it is fully functional without any errors and satisfies all the requirements of the customers. It will lead to the convenience of the customers and make our application user-friendly.

#### If the customer wants us to add a feature to rent a cab so that they can drive it by themselves instead of hiring a driver then we will make changes to our application according to their needs so that the end-product will satisfy all their requirements.

## Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.

#### In order to maintain a constant rate of development a repeatable and iterative pattern will be established so that the developers associated with our project will be not overwhelmed and over-stressed by the work load, while the project will keep on progressing at an acceptable rate. Our developers will have to develop the features of our application in a strategic and fixed pattern and will get sufficient amount of time to complete their job so that they does not get overwhelmed by work-load.

#### The Agile methodology aims at keeping the perfect work-life balance and never over exhaust the employees, thus keeping them happy. Thus, by following it our developers will give their maximum efforts during the working hours which will ultimately benefit our project.

## Continuous attention to technical excellence and good design enhances agility.

#### Providing value to the customer is the primary objective of any Agile team. To complete our project, we will hire a multi-skilled team that will handle all the technical aspects and provide the opportunity for continuous improvement. The team will constantly provide their input to improve the functionality of the application which would result in a satisfactory end-product.

#### Self-organising teams are the key to yield the best architectures, designs, and requirements. The team engages in retrospective meetings that hold discussions on the things needed in order to be more effective.

## Simplicity-the art of maximizing the amount of work not done is essential.

Simplicity is the key to success. The project should be simple and understandable during the development stage. Only the primary features should be added during the development. Too much of planning and adding extra features at the time of development will confuse the developers and will cause delay and unwanted errors. Focus should be on the things that are important to add value to the project and customers. Ignore the things that do not add value, such as components, process, etc.

For example, our application will only provide services like history of all the tourist places, cabs and hotel services at the initial stage in order to avoid confusion. Along with time various new features will be added like restaurants, online booking tickets, navigation etc.

## The best architectures, requirements, and designs emerge from self-organizing teams.

#### To develop our application, all the self-organized teams will have decision making powers so that it will reduce the responsibility from the shoulders of the project manager. As all the members of the team are allowed to tackle with a problem related to their field of expertise , it will be dealt with a faster pace and more efficiently as the decision making does not depends on a single person.

## At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behaviour accordingly.

Agile methodologies stand on the concept of iteration, where teams learn from their past mistakes and continuously improve their performance. To improve their skills and efficiency we will conduct sessions where they can reflect upon their past performances and discuss ideas to improve their skills and productivity. This will drastically improve the functionality of our project and the customers will be satisfied with our work.

# Software characteristics

## 

## Functionality

Functionality is the essential purpose of any product or service. for Certain items this is relatively easy to define, for example – In our SAFAR application so far should also provide Some functions like: - Available Hotel, Cab services, history of place etc.

Suitability- This is the essential functionality characteristics and refers to the appropriateness of the functions of the software i.e. history.

Accurateness - This refers to the correctness of the function. Ex – ‘SAFAR’ provide accurate details of hotels in budget.

Interoperability-Means capability of the software to interact with one or more specified systems, it can interact with other components or system.

Compliance- Where appropriate curtain industry law and guidelines need to be Compiled with.

Security- This sub characteristics relates to unauthorized access to the software function. Our software capable to prevent unintended access and resist deliberate attacks intended to gain unauthorized access to confidential information.

## Reliability

#### Once a software system is functioning, as specified, and delivered the reliability characteristic defines the capability of the system to maintain its service provision under defined conditions for defined periods of time. For example- if the network goes down for 20 seconds then comes back the system should be able to recover and continue functioning.

#### Recoverability- Ability to bring back a failed system to full operation, including data and network connections.

#### Fault tolerance- The ability of software to withstand (and recover) from component, or environmental, failure our software maintain a specified level of performance in case of software faults.

#### Maturity- Our software is capable to avoid failure as a result of faults in the software so that customer doesn’t face any problem.

## Efficiency

#### This characteristic is concerned with the resources used when providing the required functionality. The amount of space, memory, network etc. provides a good indication of this characteristic.

#### Time behaviour- This software is capable to provide appropriate response or result and processing times and throughput rates when performing its function under stated conditions. For ex-If user search for hotels in given location, it gives result under stated condition in less time.

#### Resource behaviour- Characterizes resources used, i.e. memory, CPU and network used when the user use to perform any function like searching history of any historical place or at the time of searching hotels.

## Usability

#### Usability only exists with regards to functionality ease use for a given function for example a function of a travelling provides sorted hotels as requested. Understandability determines the ease of which the systems functions can be understood, relates to user mental models in Human Computer Interaction methods.

#### Learnability-Learning effort for different users, i.e. our software SAFAR enable the user to learn its application.

#### Operability-Our software is easily operated by a given user in a given environment. i.e:- it is easy to use.

## maintainability

#### The ability to identify and fix a fault within a software component is what the maintainability characteristic addresses. Anything that helps with identifying the cause of a fault and then fixing the fault is the concern of maintainability. Also the ability to verify (or test) a system, i.e. we can test our software time to time to fix any bugs.

#### Testability - Characterizes the effort needed to verify (test) a system change. ex- if in future we make any change in our software, i.e- if we want to add nearby restaurants, we need to test this change.

#### Stability-The capability of the software to minimize unexpected effects from modifications of the software like as we discussed before for adding new features.

#### Changeability-Characterizes the amount of effort to change a system. Our software is enable changes like cab service to rental cab to be implemented.

#### Operability-Characteristics the ability to identify the root cause of a failure within the software.

## Portability

This characteristic refers to how well the software can adopt to changes in its environment or with its requirements. i.e – our software is portable to multiple operating system.

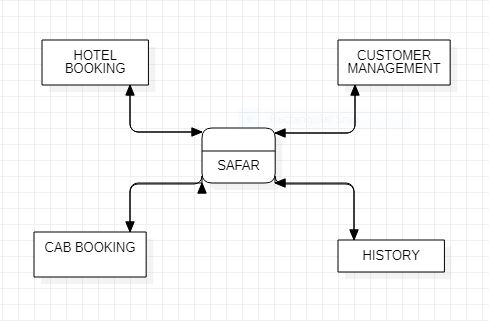
#### Adaptability-Characterizes the ability of the software to change new specifications like online ticket booking for new use or purpose.

#### Instability-The capability of the software to be installed in a specified environment like IOS, Android and Windows.

#### Replaceability-The capability of the software to be used in place of other specified software in the environment of that software. For example – in the place of different-different software SAFAR is solution for traveller.

# Data flow diagram

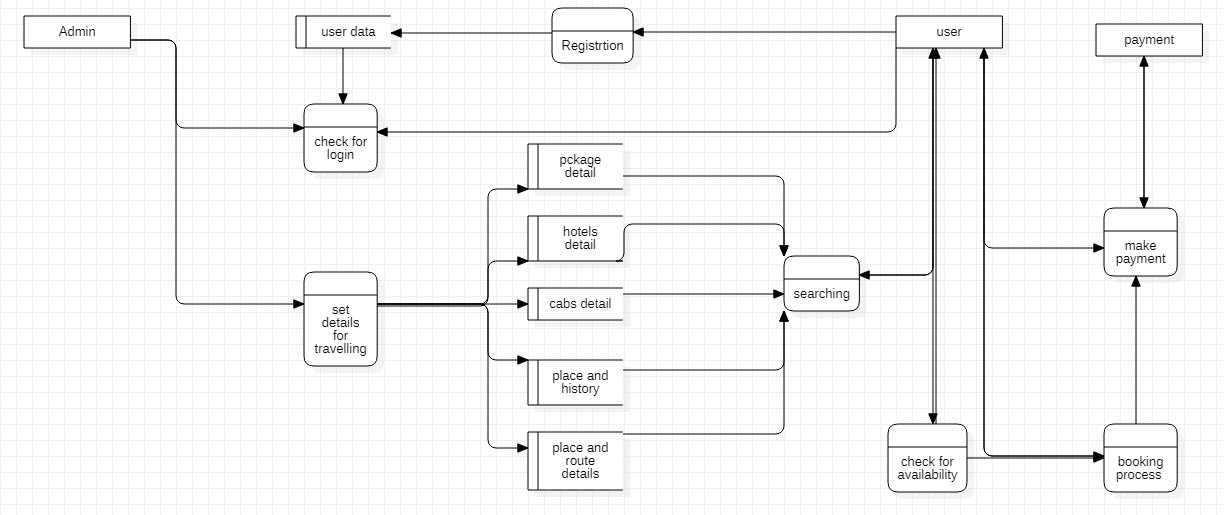
## Level 0



## Level 1

#### 

## Leve 2



# ER diagram

|  |
| --- |
| **Permission class** |
| +permission\_tittle: string  +permission\_module: string  +permission\_description: string |
| +Add\_permission()  +edit\_permission()  +delete\_permission()  +search\_permission() |

|  |
| --- |
| **Customer class** |
| +customer\_id: int  +customer\_name: string  +customer\_mobile: string  +customer\_email: string  +customer\_address:string  +customer\_username: string  +customer\_password: string |
| +addCustomer( )  +editCustomer( )  +deleteCustomer( )  +searchCustomer( ) |

|  |
| --- |
| Class diagram |

|  |
| --- |
| **Booking class** |
| +booking\_id: int  +booking\_type: string  +booking\_description: string  +booking\_title: string  +booking\_hotel\_id: int  +booking\_date: date |
| +addBooking( )  +editBooking( )  +deleteBooking( )  +searchBooking( ) |

|  |
| --- |
| **User class** |
| +user\_id: int  +user\_name: string  +user\_email: string  +user\_dob: date  +user\_address: string |
| +addUser( )  +editUser( )  +deleteUser( )  +searchUser( ) |

|  |
| --- |
| **Place&history class** |
| +place\_name: string  +place\_location: string  +place\_type: string  +place\_description: string |
| +addPlace( )  +editPlace( )  +deleteplace( )  +searchPlace( ) |

|  |
| --- |
| **Package class** |
| +package\_id: int  +package\_name: string  +package description: string  +package\_type: string  +package \_amount: string |
| +addpackage( )  +editPackage( )  +deletePackage( )  +searchPackage( ) |

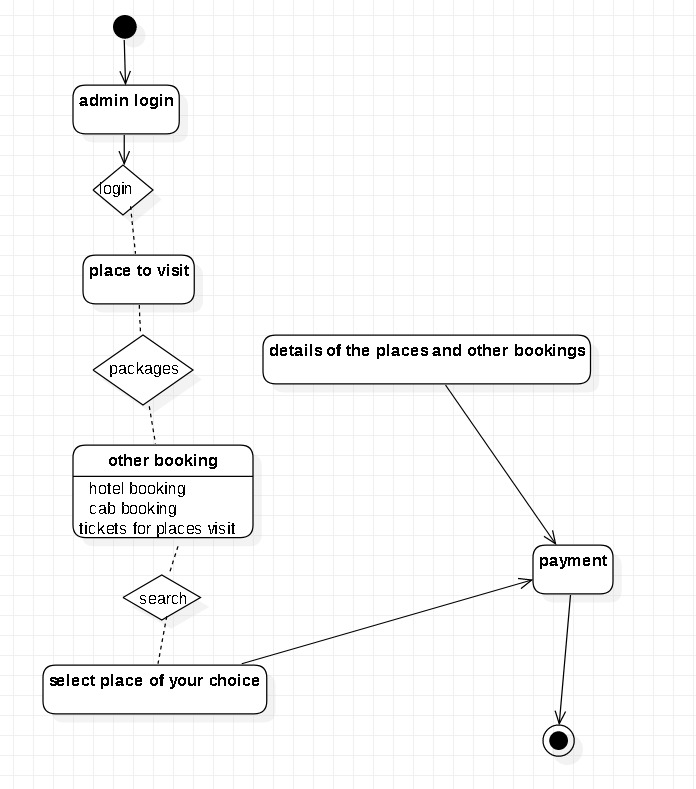
|  |
| --- |
| **Hotel Class** |
| +hotel\_id: int  +hotel\_address: string  +hotel\_rent: string  +hotel\_name: string  +hotel\_type: string  +hotel\_description: string |
| +addhotel( )  +edithotel( )  +deletehotel( )  +searchhotel( ) |

|  |
| --- |
| **Cab class** |
| +cab\_name: string  +cab\_id: int  +cab\_type: string  +cab\_descrition: string  +cab\_tour\_id: int |
| +addcab( )  +editcab( )  +deletecab( )  +searchCab( ) |

# Sequence diagram

# 

# State chart



# Scrum methodology

Scrum is an agile project management methodology or framework used primarily for software development projects with the goal of delivering new software capability every 2-4 weeks. It is one of the approaches that influenced the [Agile Manifesto](http://agilemanifesto.org/), which articulates a set of values and principles to guide decisions on how to develop higher-quality software faster.

**Scrum is a sub-group of agile:**

Agile is a set of values and principles that describe a group's day-to-day interactions and activities. Agile itself is not prescriptive or specific.

The Scrum methodology follows the values and principles of agile, but includes further definitions and specifications, especially regarding certain software development practices.

**Benefit**

* Higher productivity
* Better-quality products
* Reduced time to market
* Improved stakeholder satisfaction
* Better team dynamics

## The Components of Agile Scrum Development.

The Scrum methodology is defined by team roles, events (ceremonies), artifacts.

### The Scrum Team

Scrum teams are typically composed of 7 +/- 2 members and have no team leader to delegate tasks or decide how a problem is solved. In the development of our applications our scrum team as a unit will decide how to address issues and solve problems faced by them. Each member of the Scrum team will be an integral part of the solution and will be expected to carry a product from inception to completion. There are three key roles in a Scrum team:

**The Product Owner**

The product owner in the case of our application is Souvik Bardhan. He will be the project's key stakeholder - usually an internal or external customer, or a spokesperson for the customer. There is only one Product Owner and he will be responsible to convey the overall mission and vision of the product which the team is building. He will be ultimately accountable for managing the product backlog and accepting completed increments of work. Our product will be made according to his vision and any changes to be made will be conveyed by him to the Scrum Master.

**The ScrumMaster**

Shashank Shekhar is our Scrum Master. He will act as the servant leader to the Product Owner, Development Team and Organization. He will not have hierarchial authority over the team but rather more of a facilitator, the ScrumMaster ensures that the team adheres to Scrum theory, practices, and rules. The ScrumMaster protects the team by doing anything possible to help the team perform at the highest level. This may include removing impediments, facilitating meetings, and helping the Product Owner groom the backlog. Any changes that the product owner would make will be conveyed to him. He acts as an organizer or manager of the Scrum team.

**The Development Team**

Richa Yadav is part of the Scrum team. The Development Team is a self-organizing, cross-functional group armed with all of the skills to deliver shippable increments at the completion of each sprint. Scrum broadens the definition of the term "developer" beyond programmers to include anyone who participates in the creation of the delivered increment. There are no titles in the Development Team and no one, including the ScrumMaster, tells the Development Team how to turn product backlog items into potentially shippable increments. She will be free to deal with the problems in a way she feels comfortable as there is no one to order the scrum team. They are individual entity and has a right to provide input for our application Safar.

## **Scrum Events (Ceremonies)**

The Sprint

A sprint is a time-boxed period during which specific work is completed and made ready for review. Sprints are usually 2-4 weeks long but can be as short as one week. Example- we can make a sprint such as for developing process of history and booking tickets for which we can take a week time to complete and then present it.

Sprint Planning Sprint

Planning team meetings are time-boxed events that determine which product backlog items will be delivered and how the work will be achieved. In this we will be planning how the process will be taking places such as how we will be developing the website the design of the website .

The Daily Stand-up

The Daily Stand-up is a short communication meeting (no more than 15 minutes) in which each team member quickly and transparently covers progress since the last stand-up, planned work before the next meeting, and any impediments that may be blocking his or her progress.in this meeting we usually discuss how much work each of the team member has done after the work is been allotted to them .

The Sprint Review

The Sprint Review is the "show-and-tell" or demonstration event for the team to present the work completed during the sprint. The Product Owner checks the work against pre-defined acceptance criteria and either accepts or rejects the work. The stakeholders or clients give feedback to ensure that the delivered increment met the business need. For this sprint reviewing in this process the development team will be showing the sprint which they have completed such as when the sprint of history and booking tickets will be showing to the product owner if he thinks it is good enough for which the users will be happy using it and if he does not like the idea then he can reject it.

The Retrospective

The Retrospective, or Retro, is the final team meeting in the Sprint to determine what went well, what didn't go well, and how the team can improve in the next Sprint. Attended by the team and the ScrumMaster, the Retrospective is an important opportunity for the team to focus on its overall performance and identify strategies for continuous improvement on its processes.This is the last meeting where will be discussing about all the sprint and finally deciding what is good for the users to use in the end and we can also improve the sprint which are not up to mark .

### Scrum Artifacts

Product Backlog

Product backlog is ordered list of tasks and requirements the final product actually need.It is constantly evolving and never complete the product owner oversees the product backlog including how its made available to the team its content how its ordered and everything so the product owner and the rest of the team work together to review the product backlog and make adjustments whenever necessary as product requirement keep changing throughout the lifecycle so like agile is a night recitative approach so scrum so the product owner and the rest of the team keep updating product backlog according to the requirement that keep changing during different iteration so for each item in the product backlog we should add some additional information like hotel description, history of the place online tickets as in the priority order within which the whole backlog is listed the estimate time and the value it has in the business apart from this we can add many other things like cab description, bus service restaurants etc.

Sprint Backlog

A sprint backlog is the specific list of items taken from the product backlog which are to be completed in a sprint so team member sign up for the next tasks of safar in the sprint backlog on their skills and priorities so as the sprint the vols it is possible that slight change to the sprint backlog might occur a new understanding feature of our product like payment, and customer review the team is working on could mean that something is added to or removed from the sprint backlog so this print backlog is real time picture of the work the team currently plans to complete during a sprint. For example if the product owner needs to replace cab services and add self driving vehicle.etc

Product Increment

An Increment is the sum of all product backlog items that have been completed since the last software release. While it is up to the Product Owner to decide on when an increment is released, it is the team's responsibility to make sure everything that is included in an increment is ready to be released. In our product safar increment is released with only basic features like online tourist guider and online tickets of fun places where tourist want to visit. This is also referred to as the Potentially Shippable Increment (PSI).

**The Burndown Chart**

Burndown charts are graphs that give an overview of progress over time while completing a project. As tasks are completed, the graph “burns down” to zero. It is used as a tool to guide the development team to a successful completion of a Sprint on time with a working final product. If a team decides they have moved more objectives than possible for completion from the Product Backlog to the Sprint Backlog, the Burndown Chart can aid them is ascertaining which tasks they are not realistically able to complete so that these task can be moved back to the Product Backlog.