Introduction

As mentioned in the proposal we are going to develop a system that is going to be used to book, ticket and view available information’s about the newly emerging long distance train in Ethiopia.

And this documentation is going to include the description of the features the system is going to include. In addition it is going to include the highlight description of how the system is going behave with a certain users and a certain criteria’s.

This documentation will also include the functional and non-functional requirements of the system. Like also mentioned in the above statements it will show the interaction of the user to these requirements. And also it will tell or define how the system will work based on these requirements.

To conclude, this documentation is about the systems specific requirements. Which generally describes the interaction of the user and the system and how they behave statically with out detailed data definition.

Background of the system

In this section we are going to discuss the previous working system and how the interaction was with the users with the given system.

We developed our system because the we believe there is a huge wastage of both the staff’s resource and the user’s resource as well. And our proposed system will solve some of the problems that the current working system will provide.

So, when we come back to how the current working system works is that, Most of the functionalities are done manually. And users must arrive at ticketing stations to do their bidding.

These requirements are gathered first hand from the working staff of the organizations. And some of these requirements are coming from some of the users perspectives who have done things in real life. Meaning the users have to come to the offices manually and get information and get their services.

Data management was done manually or done partially done with partially done with the help of computer programs like excel and other programs.

The user’s access to the system is limited to the ticketing offices. There was no remote access. There were no mechanisms of accessing the services provided by the organization.

These all problems are making the life of the users uneasy and the organizations data management is very unorganized. So, the systems we are providing manages the problems that are provided above.

General and specific Objectives

The objectives of the system we are going to develop will solve the given problems of the current working environment. Those objectives can be classified in specific and general objectives.

The specific objectives: - are objectives that solve a specific problems that the organization and users provide. Eg :- book tickets.

The specific objectives provide the main functionalities that the system provide. This doesn’t mean the that the general objectives aren’t main functionalities, they just are view of the major perspective. Or General objectives provide with the bigger picture.

Here are few general objectives of our system.

* Automation
* Easy data management
* Availability
* Effectiveness and efficiency
* Resource management

And some of the specific objectives are listed down below:-

* Ticket management and reservation
* Report generation
* Security
* Manage payment activities
* Provide extra information on destination cities
* Provide Information(history and new news)
* Employee management

These are the basics but through time and during code implementation this objectives might get broader even there might be additional objectives that the system will uphold.

Other features will be discussed later in the functional and non functional requirements are defined.

How the system works

In this part we will discuss briefly how the system is going to work and how many actors there are. And Also, we are going to see if our system will interact with external environment.

To start in general manner, the system will have three parts. The first part there is web application for the user. This web app consists of different features like book ticket, book and pay ticket, view travel information, manage seat information, choose class, and other ticket management issues for the long distance Ethiopian rail way.

This web app doesn’t require any login and sign up formats because any user can access it and use it for their needed purpose.

The second system is still again a web but this web is developed for the admin and staff workers of the organization. And this manages the user activity both on their remote activities and manual transactions in the ticketing offices.

The admins manage both the staff workers(employee management) and also manages the users interactions with the staffs. But the user is mainly interactive with staff members when they arrive to their respective offices.

The admins have privileges to see the individual staff members activity. Manages the accounts of the employees. And able to see reports generated automatically by the system.

The staff members are able to reserve tickets, convert booked tickets to reserved tickets, cancel reserved tickets, change reservation ticket information, view available information’s about future tickets, and other features that might be added through time. To do all this activities all staff members and admins must login.

We are implying that there might be additional features is that because we are using agile software development methodology. And this technique will help us add up new features to the system through the development process.

The third system is a mobile application for the users. It has mostly the same features as the web application which is implied for the users but with a little bit more added features.

This mobile application increases the mobility of the system in a big way. Because of this mobile app a lot of users will become more eligible for this system.

As the technology increases the want for big system to be in small device is increasing dramatically, our system will fulfill these wants. And also gains more attraction from the society.

There are interactions with external systems for the sake of few reasons. One of the reasons is, there might be some needed information about services that are available near our destination areas and there might be links that forward us to their pages and systems.

The other reason is we are asked to develop a payment mechanism that is going to be integrated with our system. so these payment system needs to be externally integrated with our system like an API. And we might use API’s that are provided by online banking systems. Or a simulation web services that will act us our payment mechanism.

Functional and non-functional requirements

To identify the functional and non-functional requirement we used the following criteria’s:

* The functional requirements are the requirements that construct the basis of the system
* Functional requirements are the requirements that we get directly from the stakeholders required functionalities.
* The non-functional requirements are the requirements that we gather indirectly from the stake holders.

If we say the following concepts lets get to the listing of those requirements.

Functional requirements

* Book a ticket - login
* Reserve ticket - cancel travels/reservations
* View available sites - change reservation information
* View available travels - view previous trips
* Pay for booked tickets - view reports
* Add new travel information’s - add and remove employees

Most of the functional requirements are listed above. The non-functional requirements are:

* Availability
* Mobility
* Easy access
* Understandability
* Fast and reliable
* Security
* Maintainability
* Reusability
* Scalability
* Lower down time

Models

In this paper there are going to be three models that are used to represent the structure of the system.

The first model is use case model this model represents each and every interactions of the users to the functionalities they access.

This model also defines and describes about the use cases. Meaning it gives brief description about each uses cases like who initiates them other actions they take, flow of events alternative events and others.

The other model is the class model. This defines the data structure of the system like, the classes it hold and attributes it consists of and the methods it holds.

This model is very necessary and mandatory for developing the code aspect and also to create the database of the system.

The last but not the least model used to represent our system in this paper is deployment model.

This model shows the placement of components or devices while the system is running. This model helps us to understand the working mechanism of the devices.

Use case model

Actors

* User(can be using both web and mobile app portals)
* Staff member
* Admin

Use cases for each users

Users

* View available travel info
* Book a ticket
* Reserve a ticket
* Pay for ticket
* View available extra features

Staff members

* Pay booked tickets for users
* Reserve tickets for users
* Insert user info
* View available travel information
* Cancel reserved tickets
* Change reserved ticket info
* Login
* Log out

Admin

* Inset news
* Login
* Logout
* View reports
* Add staff members
* Remove staff members

Use case modeling

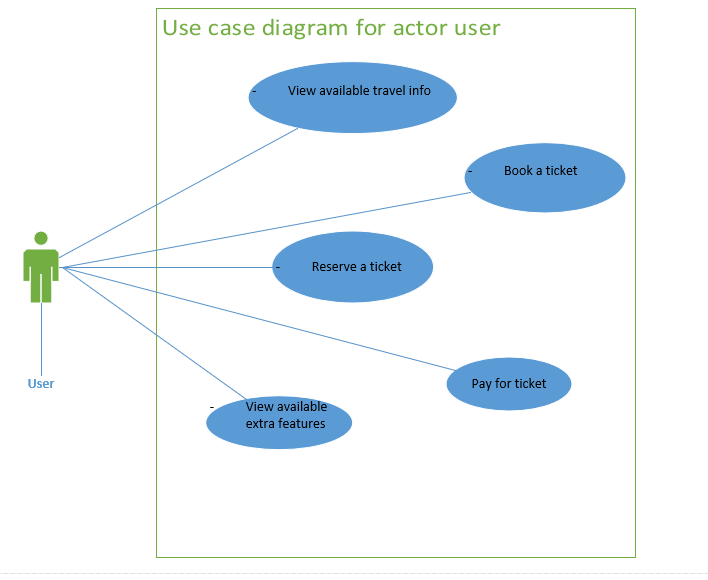


Figure 1.1 use case diagram for user actor

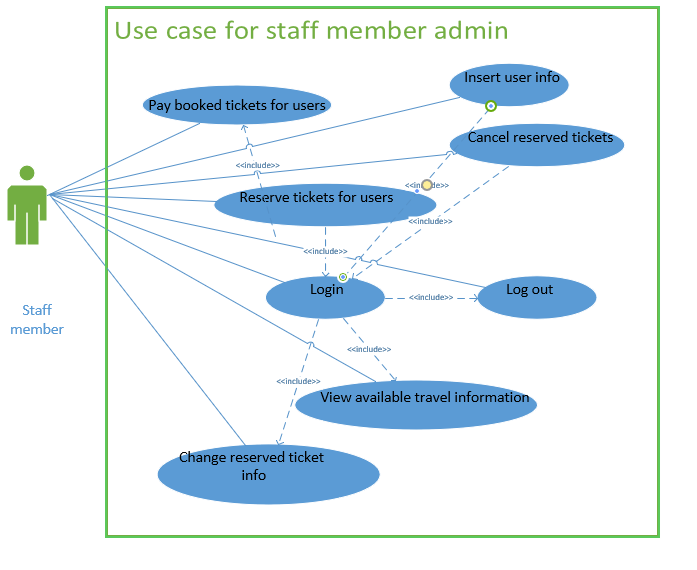


Figure 1.2 use case diagram for staff members

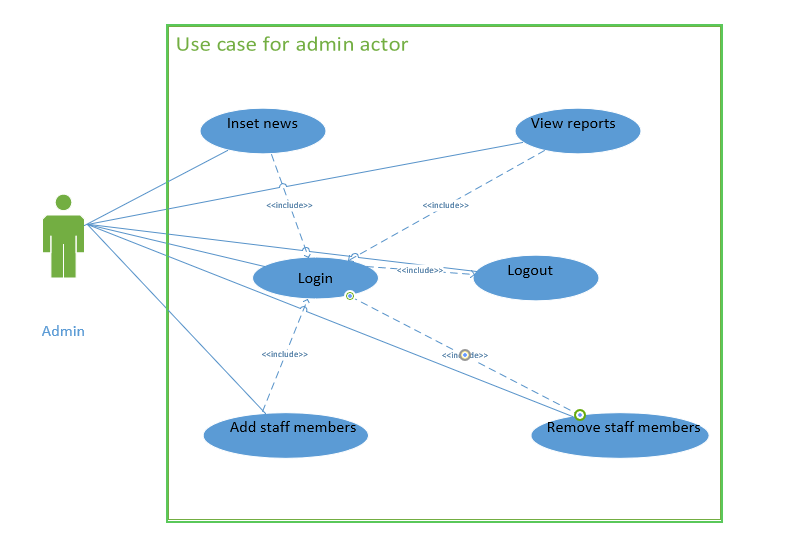


Figure 1.3 usecase diagram for admin actor

Use case descriptions

|  |  |
| --- | --- |
| Use case id | UC1 |
| Name | Book a ticket |
| Description | Books a ticket of a travel for later use |
| Actor | user |
| Pre-condition | Should select a single travel information from the available travel information |
| Post-condition | Booked ticket must be paid for the ticket to get justified |
| Main success scenario | 1. The user accesses a web or mobile app portal 2. The user fills the necessary form 3. The system validates the form and checks for available travels on that inserted information 4. The system displays booked information to user for validation. 5. The user accepts the displayed information. 6. The system saves the inserted data to the Database for later use. |
| Alternative scenario | 4.1 the system sends back no available travels on that specified information.  4.2 the user reinserts the form to check available travels. |

|  |  |
| --- | --- |
| Use case id | UC2 |
| Name | View available travel info |
| Description | Displays available travel information |
| Actor | user |
| Pre-condition | none |
| Post-condition | Make information available to book and reserve a ticket |
| Main success scenario | 1. The user accesses a web or mobile app portal 2. The user fills the necessary form 3. The system validates the form and checks for available travels on that inserted information 4. The system displays available information to the user. 5. The user selects the displayed information. 6. The system saves goes to booking or reserving pages. |
| Alternative scenario | 6.1 user can cancel and get back to home and other pages. |

|  |  |
| --- | --- |
| Use case id | UC3 |
| Name | Reserve a ticket |
| Description | To reserve new ticket for travel information |
| Actor | User and staff |
| Pre-condition | If staff must login. |
| Post-condition | Ticket number and another user information’s are displayed and stored. |
| Main success scenario | 1. The actor requests to reserve a ticket 2. The system provides the necessary form to be filled. 3. Actor fills those forms then submits 4. The system verifies and send a confirmation message and requests user to go to payment mechanisms 5. The actor then go to payment mechanism. |
| Alternative scenario | 4. The system may send error messages if the filled information was incorrect. |

|  |  |
| --- | --- |
| Use case id | UC4 |
| Name | Pay for ticket |
| Description | Payment for booked or reserved ticket |
| Actor | User and staff |
| Pre-condition | For payment of booked ticket user must have booked a ticket using web or mobile app portal.  if staff must login. |
| Post-condition | Ticket number and another user information’s are displayed and stored. |
| Main success scenario | 1. The necessary actor initiates the payment mechanism.   1. The Actor Chooses from booked ticket and reserved ticket. 2. If booked is chosen the system provides form to fill about booking information   3.1 The actor fills booking information and submits.  3. 2 The system checks the input values and validates and sends the confirmation message.  3.3 The actor gets the full travel information and gets print out.   1. The system accepts the reservation information and provides form to fill about payment information   4.1 the system verifies and send confirmation.  4.2 The Actor get the full travel information and gets print out. |
| Alternative scenario | 3.2 and 4.1 The system will send error message and get back to the page if any wrong information is inserted. |

|  |  |
| --- | --- |
| Use case id | UC5 |
| Name | Cancel reserved ticket |
| Description | When a user needs to cancel there travel plan and the staff member executes there wish |
| Actor | Staff |
| Pre-condition | staff must login and user must have a reserved ticket |
| Post-condition | Ticket will be cancelled. |
| Main success scenario | 1. The system provides for the insertion of ticket number. 2. The staff member inserts the users ticket number. 3. The system checks if the inserted information is correct 4. Then the system will calculate the punishment fee and will show that and the return fee. 5. The staff member accepts and proceeds. |
| Alternative scenario | 3. The system might return error information like the booking number not correct or the booking has expired. |

|  |  |
| --- | --- |
| Use case id | UC6 |
| Name | change reserved ticket info |
| Description | When a user needs to change there travel plan and the staff member executes there wish |
| Actor | Staff |
| Pre-condition | staff must login and user must have a reserved ticket |
| Post-condition | Ticket will be changed to new travel plan. |
| Main success scenario | 1. The system provides for the insertion of ticket number. 2. The staff member inserts the users ticket number. 3. The system checks if the inserted information is correct and provides a form for the new travel plan. 4. The staff member fills the form and proceeds 5. Then the system will calculate the punishment fee and will show that and the new fee. 6. The staff member accepts and proceeds. |
| Alternative scenario | 3. The system might return error information like the booking number not correct or the booking has expired. |

|  |  |
| --- | --- |
| Use case id | UC7 |
| Name | Pay booked tickets for users |
| Description | Pay and reserve the ticket of users for later use |
| Actor | staff |
| Pre-condition | staff must login and user must book a ticket |
| Post-condition | Ticket will be reserved and be available to use. |
| Main success scenario | 1. The system provides for the insertion of booking number 2. The staff member inserts the user booking information 3. The system checks if the inserted information is correct 4. Then the system will display the users information and request to proceed to payment. 5. The staff member selects h proceed button. 6. The system will provide a form for payment information. 7. The staff member will fill the payment information. 8. The system will display all information and prints out the ticket. |
| Alternative scenario | 3. The system might return error information like the booking number not correct or the booking has expired. |

Class model

This model is used to reperesent the dta structure of the system and the below diagrams and modeling concepts reperesent our system data concept.

The classes of our systems are: -

* Staff: - this class stores and manages the information about staff employees and admins.
* Users: - this class is modeled to manage and handle user informations
* Reservation: - is the class that is created to manage the events that are related to reservations.
* Payment: - this class holds and manages information related to payment mechanisms
* Routes: - this like the above class manages its own attributes and methods which are related to there own features, which in this case is route infromations.

Relationships

* A reservation can only have one payment mechanism.
* A reservation has only one route information.
* One user can make multiple reservations.
* One staff member can make multiple reservations.

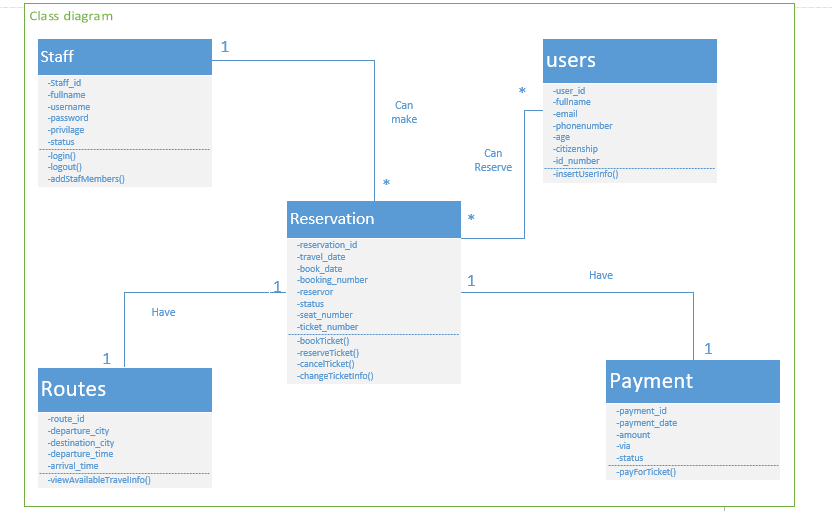


Figure 2.1 class diagram