



FINAL SEMESTER ASSESSMENT (FSA)
B.TECH. (CSE)
SEMESTER VI

**UE18CS355 – OBJECT ORIENTED ANALYSIS AND DESIGN
WITH SOFTWARE ENGINEERING LABORATORY**

PROJECT REPORT ON
BUS TICKETING SYSTEM(VAHANA)

SUBMITTED BY

NAME

SRN

1) Teja Puvvadi	PES2201800163
2) Goutham Swaminathan	PES2201800042
3) Keshav Shivkumar	PES2201800168

JANUARY – MAY 2021
DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
EC CAMPUS
BENGALURU – 560100, KARNATAKA, INDIA

TABLE OF CONTENTS

Sl. No	TOPIC	PAGE #
1.	ABSTRACT	3
2.	SOFTWARE REQUIREMENTS SPECIFICATION	3
3.	PROJECT PLAN	10
4.	DESIGN DIAGRAMS	14
5.	MODULE DESCRIPTION	21
6.	TEST CASES	21
7.	FINAL OUTCOME(SCREENSHOTS)	29

Abstract

This project aims to utilize principles of software engineering to generate a commercially viable product. Our project, Vahana, is a bus ticketing website that is designed in the vein of other prominent bus booking applications, like RedBus. E-ticketing in India has seen a boom in the last 10 years and has become a rather popular tool for booking tickets. The rise can be attributed to the bus industry in this country being highly fragmented and the lack in organisation. Another reason is because other tickets are costlier and train tickets have to be booked weeks before, making the booking of bus tickets the practical option for long distance travel.

The Vahana platform will only accept certified bus operators to ply their buses through the website. The user is expected to be able to browse the list of desired buses and book, for which he/she will get tickets. The user has the advantage of being able to cancel any booking he/she may not want anymore. Moreover, bus operators also have the benefit of signing up and logging in, and can handle their organization's bus division. They can add new buses and remove buses from service as per their needs.

A bus ticketing application involves making it easy for the user to book a bus to go to an area online. Making an effective bus ticketing application would necessitate ensuring a simple, safe and reliable interface to book without any setbacks and issues. Basic functional requirements such as searching, booking, cancelling, etc. must be covered through a clean and easy-to-use interface.

Software Requirements Specification

1. System Features

1.1 Login/Sign up

These two features are quintessential in any e-commerce application, since these define a user's legitimacy and help the Bus Operators identify who is buying their products. If there wasn't such a system in place, there would be questions in the legitimacy of the user and their methods of purchase. Besides users being able to login/sign up, a bus operator has the advantage to login/sign up as well.

If the user does not own an account, he/she clicks on the "sign up" button. Here, they enter their account details (name, email, password, etc.). After entering the required details, they click the "create account" button. An account is created for the user, and the user is redirected to the login page.

In the login page, registered users can enter the required details. If an operator is logging in/signing up, there is a checkbox to tick.

If these match with the database, then they're directed to their dashboard.

Otherwise, an error message pops up saying that the details entered are incorrect.

FUNCTIONAL REQUIREMENTS:

- A system that supports Python3.
- A database that holds the details of all users and products.

1.2 Searching for buses

Booking a desired bus clearly involves searching for one. The user must be able to search for the bus that takes him/her to the required destination.

Being able to search for a bus is a key functionality that can't be ignored.

Logging in is not required to search for buses. A person can fill in his requirements for the bus he/she is searching for, and get a list of available buses. For example, the person can search for all buses that drive to a certain destination on a certain date, and get all the available buses that travel to that destination. To book a bus, however, logging in is required.

FUNCTIONAL REQUIREMENTS:

- A system that can run Python3.
- A database to store data

1.3 Viewing available buses

Unregistered users will be able to look up the buses plying that they desire, but can't buy them because they don't have an account registered on the application. Consider these types of users as window shoppers, they can look at the buses they want to travel in all they want, but in order to purchase said item, they have to get it billed from one of the employees in the store.

The unregistered user can have a look at all the buses plying on the website's database. They can also go over the details of each bus. If this user clicks on the click to book for the bus(es) they want to travel in, they get directed to the signup page. Here, the user can either: Login to their already created account (or) Fill up the sign up form to create a new account.

FUNCTIONAL REQUIREMENTS:

- A system that can run Python3.
- A database that holds details of all users.

1.4 Booking a bus

Registered users can book the bus ticket(s) they want to buy and add to

'BOOKED'. This feature is provided so that users have a clean and specific page to book the desired bus. Unregistered users, however, will have to first create an account in order to book that ticket.

The user finds and books tickets on the bus he wants to travel in. The user clicks on 'BOOK TICKETS'. If the user is logged into his account, the ticket application gets added to the 'BOOKED' section. The user can check the 'BOOKED' section. If they want to proceed with the booking, it'll guide them to the 'CONFIRM AND PAY' section. The payment will be taken care of by a third party payment application. Once that is confirmed, the Bus Operator will be notified of the booking, and the number of seats available on that bus(es) will decrease accordingly. An unregistered user will be directed to the Signup Page.

FUNCTIONAL REQUIREMENTS:

- A system that can run Python3.
- A database to store data

1.5 Viewing booked buses

Registered users have the privilege of viewing the buses that they have booked. Besides getting a bill for booking, it is convenient for a user to see the booked buses and keep track.

To view all the buses, the user firstly must be signed in. Once a user is signed in, he/she can check a list of all buses that have been booked. There is also the option to cancel booked buses here.

FUNCTIONAL REQUIREMENTS:

- A system that supports Python3.
- A database that holds the details of all users and products.

1.6 Ticket for bus reservation

After booking a bus, the user has to get a ticket. This is crucial with respect to having a reliable economic system in the business POV. Once a bus has been booked, the user is redirected to a ticket with details, that he can print/save, and use it when boarding the bus.

Once a user logs in, he can see a button on the navigation bar called "SEE BOOKINGS". On clicking it, the user gets a list of all the buses that have been booked, along with details of each bus.

FUNCTIONAL REQUIREMENTS:

- A system that supports Python3.
- A database that holds the details of all users and products.

1.7 Cancelling a booked bus

Registered users that have already booked bus ticket(s) can choose to

cancel. This feature is provided so that users have an option for reconsideration. After making up their mind, the users can proceed to confirm a cancel for the specific bus ticket(s).

Cancelling a booked bus requires being signed in. Once a user is signed in, he/she can check a list of all buses that have been booked. The user can choose to cancel by adding the ticket ID and clicking on 'Cancel'. Once a bus has been cancelled, the user can see the status of that bus in the list as "CANCELLED".

FUNCTIONAL REQUIREMENTS:

- A system that supports Python3.
- A database that holds the details of all users and products.

1.8 Viewing buses under an operator

Bus operators can see the buses that belong to their organization. This interface is necessary so that the bus operator can handle their organization's buses without the admin of the application interfering.

Once a bus operator signs in, a list of buses under their organization, with details of each bus, can be viewed by clicking on the "CHECK BUSES" tab on the navigation bar. There are options for adding and deleting buses.

FUNCTIONAL REQUIREMENTS:

- A system that supports Python3.
- A database that holds the details of bus operators and buses.

1.9 Adding a bus under an operator

A bus operator can add buses to their organization. On adding new buses, any source and destination that is newly created gets automatically added to the list of searchable buses, thus creating a convenient website.

Once a bus operator signs in, a list of buses under their organization, with details of each bus, can be viewed by clicking on the "CHECK BUSES" tab on the navigation bar. To add a bus, the operator clicks on the "ADD BUS" button, and fills a form to add a desired bus.

FUNCTIONAL REQUIREMENTS:

- A system that supports Python3.
- A database that holds the details of bus operators and buses.

1.10 Deleting a bus under an operator

A bus operator can remove buses from their organization.

Once a bus operator signs in, a list of buses under their organization, with details of each bus, can be viewed by clicking on the "CHECK BUSES" tab on the navigation bar. To delete a bus, the operator clicks on the "DELETE BUS"

corresponding to the row of the bus that is to be deleted.

FUNCTIONAL REQUIREMENTS:

- A system that supports Python3.
- A database that holds the details of bus operators and buses.

2. External Interfaces

2.1 User Interfaces

- Register/Login/Logout
- View/Book/Cancel buses
- See all the user's booked and cancelled rides
- Bill details

2.2 Software Interfaces

- Python
- Django
- Web Browser

2.3 Communication Interfaces

- CLI
- HTTPS
- SMTP

2.4 Hardware Interfaces

- Monitor
- Keyboard
- Mouse
- Web Server

3. Assumptions and Dependencies

- Our intended crowd knows and can follow basic instructions to navigate through an application
 - Users must have access to the third party/commercial applications involved
 - General actor dependencies
 - Goal-modelling

4. Non-Functional Requirements

Performance Requirements

- Simple and easy-to-use UI
- Proper registering and logging in

Safety Requirements

- Database must be backed up periodically
- Web server must be safe from hackers

Security Requirements

- All transactions must be made securely
- Data must be secure

5. Design & Implementation Constraints

- Transactions need to be private and secure
- Unsuccessful payments need to be dealt with appropriately
- Faults on the user's end
- Possibility of data loss

Appendix A (GLOSSARY)

- Vahana: Name of our product.
- Passengers: People who want to book tickets on our application.
- Bus Operators: People/Organizations that want to sell bus tickets on our application
- Review: Feedback left by our customers after they have travelled on a bus.
- HTTPS: Hypertext Transfer Protocol Secure
- SMTP: Send Mail Transfer Protocol

Appendix B (FIELD LAYOUTS)

→ Information for the passenger to register

Field	Length	Data Type	Description	Is it mandatory?
Name	<150	String	Passenger name	Y
Email	-	String	Passenger email	Y
Password	>8	String	Password	Y
Are you an operator	-	Boolean	Ticked if the user is registering as an operator	N

→ Information for the operator to register

Field	Length	Data Type	Description	Is it mandatory?
Name	<150	String	Passenger name	Y
Email	-	String	Passenger email	Y
Password	>8	String	Password	Y
Are you an operator	-	Boolean	Ticked if the user is registering as an operator	Y

Appendix C (REQUIREMENT TRACEABILITY MATRIX)

S.no	Req ID	Description	Architecture reference	Test case ID	Test case description	Status
1	REQ 001	Register an account in the application	Two Tier	RL-0,RL-2	To test the register functionality	ALL PASS
2	REQ 002	Login to the application	Two Tier	RL-1,RL-3	To test the login functionality	ALL PASS
3	REQ-003	User searches for buses	Two Tier	US-0	To test the users ability to search	ALL PASS

4	REQ-004	User attempts to book a bus	Two Tier	US-1	To test if the user can book buses under all cases	ALL PASS
5	REQ-005	See bookings	Two tier	US-2	To test see if the user can see their booking under all cases	ALL PASS
6	REQ-006	Cancel bookings	Two Tier	US-3	To test the cancel functionality	ALL PASS
7	REQ-007	Operator views buses under them	Two Tier	OP-0	To test the operator's ability to view buses under them	ALL PASS
8	REQ-008	Operator adds buses	Two Tier	OP-1	To test the operators ability to add buses under different circumstances	ALL PASS
9	REQ-009	Operator deletes bus	Two Tier	OP-2	To test the operators ability to delete buses under different circumstances	ALL PASS

Project Plan

2.1 Development Cycle

We have chosen incremental prototyping as the development model for our project (Bus Ticketing system).

Incremental prototyping has a series of prototypes that are made, with each one identifying with a refined version that will eventually end up representing the final product.

There are a number of steps and the development happens orderly.

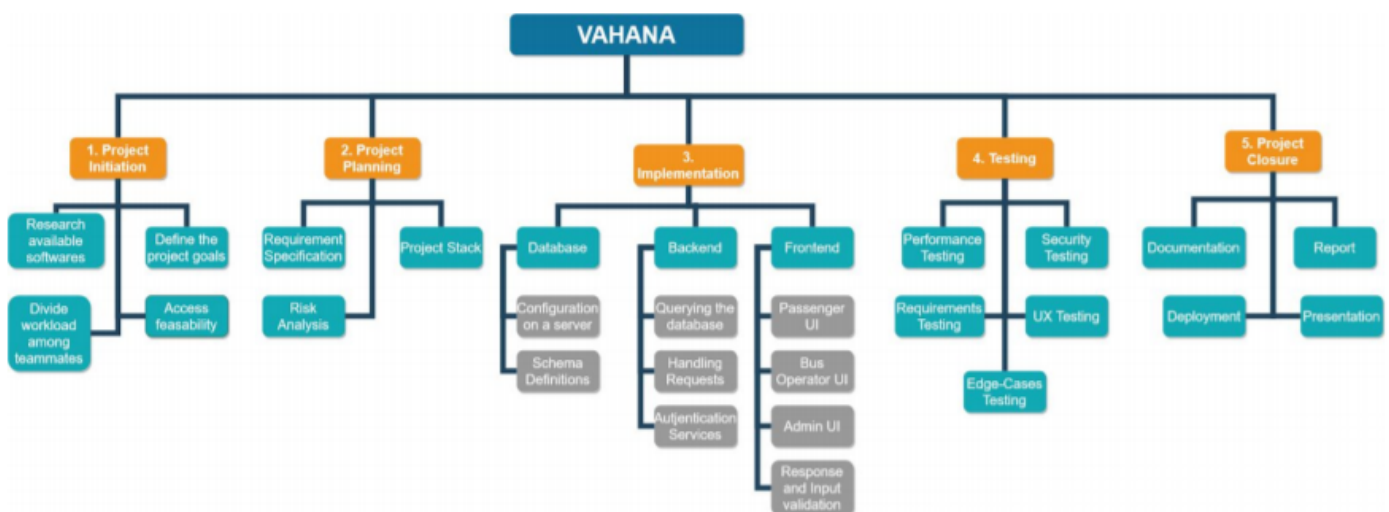
Every time there is an increment, there will be some improvement, fixing of bugs, addition of features and changes in the design itself.

This approach seemed appropriate for us because our intention was to make the software quickly and also test it thoroughly and easily. This makes sense because it is easier to test a small increment over the entire working system. This will also help in risk mitigation as issues are handled as and when an increment is made avoiding the piling up of errors and risks.

2.1.1 Advantages

- Customers can see a part of the product in early stages of the life cycle. This will make sure the customer is okay with the direction in which the product is moving.
- Any requirements that have to be made can be easily bought thanks to many incremental steps
- Functionalities are added over time so it is easy to figure out functionalities that are essential but not added yet
- We can reuse the developed prototype in the future.

2.2 Work breakdown Structure



2.3 Tools to be used

2.3.1 Planning Tools

Excel online

We use Excel online to create checklists so that we can make sure we finish all tasks.

2.3.2 Build tools

- **Django**

Django is a high-level Python web framework that enables rapid development of secure and maintainable websites. Django takes care of much of the hassle of web development. We used Django for the backend with the views.py file containing all the different views the user will be served with.

- **SQLite**

We used SQLite to store all the different tables like User, Operator as it was highly compatible with Django. (Django officially supports SQLite) Other advantages that we considered before opting for this was that it was fast, highly reliable, self contained and serverless making it optimal for our needs.

2.3.3 Version Control

Git

It is an easy-to-use platform for open-source projects. It has branching capabilities which can be used for each phase of the project.

2.4 Project Deliverables

A web application that allows users(passengers) to conveniently view bus routes plying from their desired source and destination on any given day and then proceed to login in order to book seats for their travel. They will then be presented with the ticket details, listing the associated information like total-price etc. They will be able to view all their previous, ongoing and cancelled bookings. Users can cancel their booking as they please on the site. The application will also allow another type of user (operators) to create accounts and the operators will be allowed to add a fleet of buses that will ply under the supervision of the operator. They can remove buses (added by them) from service as they please. Normal users will not be allowed to login as an operator in our application.

2.5 Effort Estimation

The estimation for efforts will be done using the Constructive Cost Model

(CoCoMo).

Since the team size is small, the problem statement is well understood and has been solved in the past, and all team members have a nominal experience in solving the problem, our project will be following the organic system type of CoCoMo.

We will be using the Basic CoCoMo model to calculate the software costs. Hence the estimations will be as follows:

- Effort = $a(\text{KLOC})^b$
- Time = $c(\text{effort})^d$
- PersonsRequired = Effort/Time

Since it is an organic project, the values of the constants are as follows:

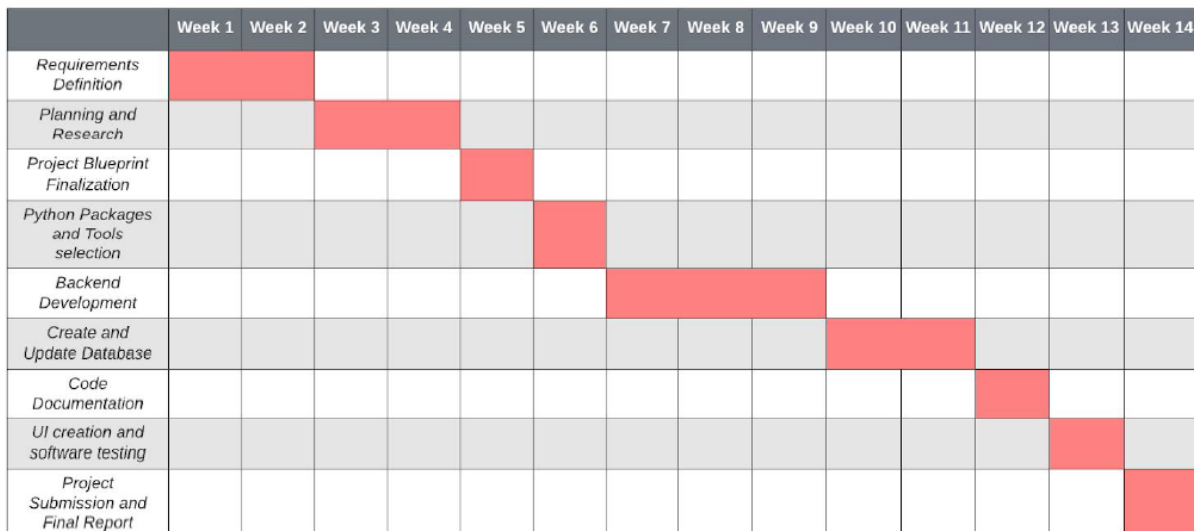
- ❖ $a = 2.4$
- ❖ $b = 1.05$
- ❖ $c = 2.5$
- ❖ $d = 0.38$

Substituting above values in the formulas, we get:

- Effort = $2.4(1)^{1.05} = 2.4$ man months
- Time = $2.5(2.4)^{0.38} = 3.48$ months
- PersonsRequired = $2.4/3.48 = 1$ person

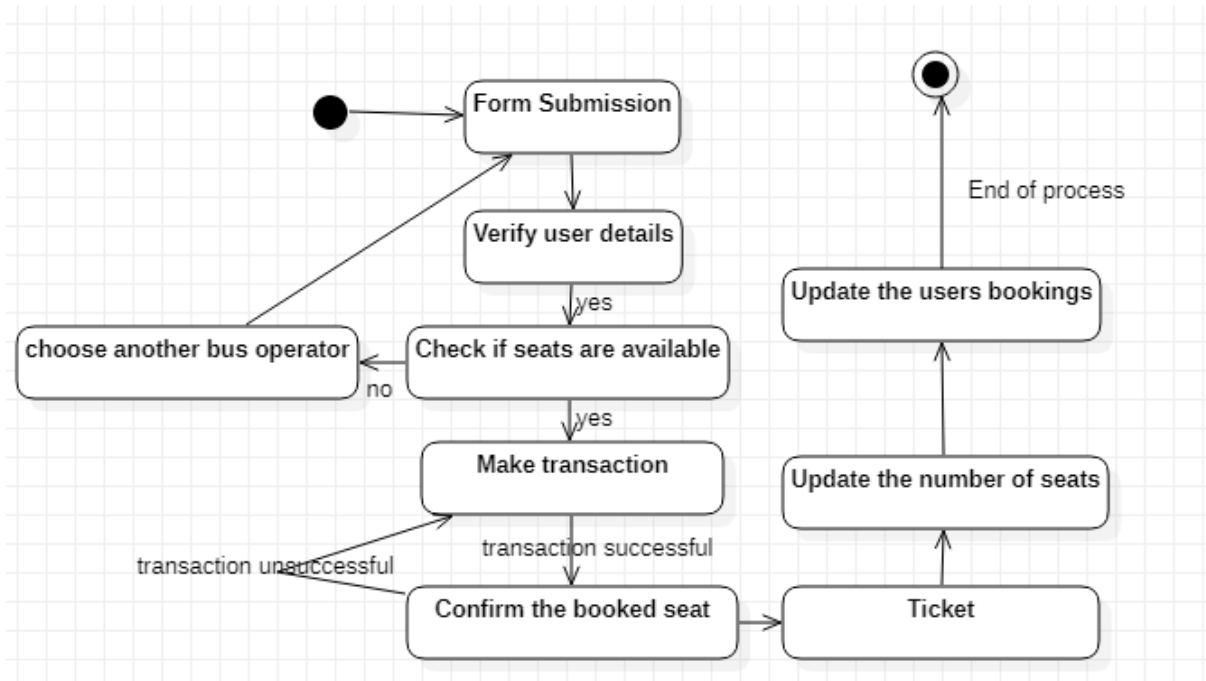
Hence 1 developer can make the project in around 3.5 months. With 3 people, it'll be done much faster.

2.6 Gantt Chart

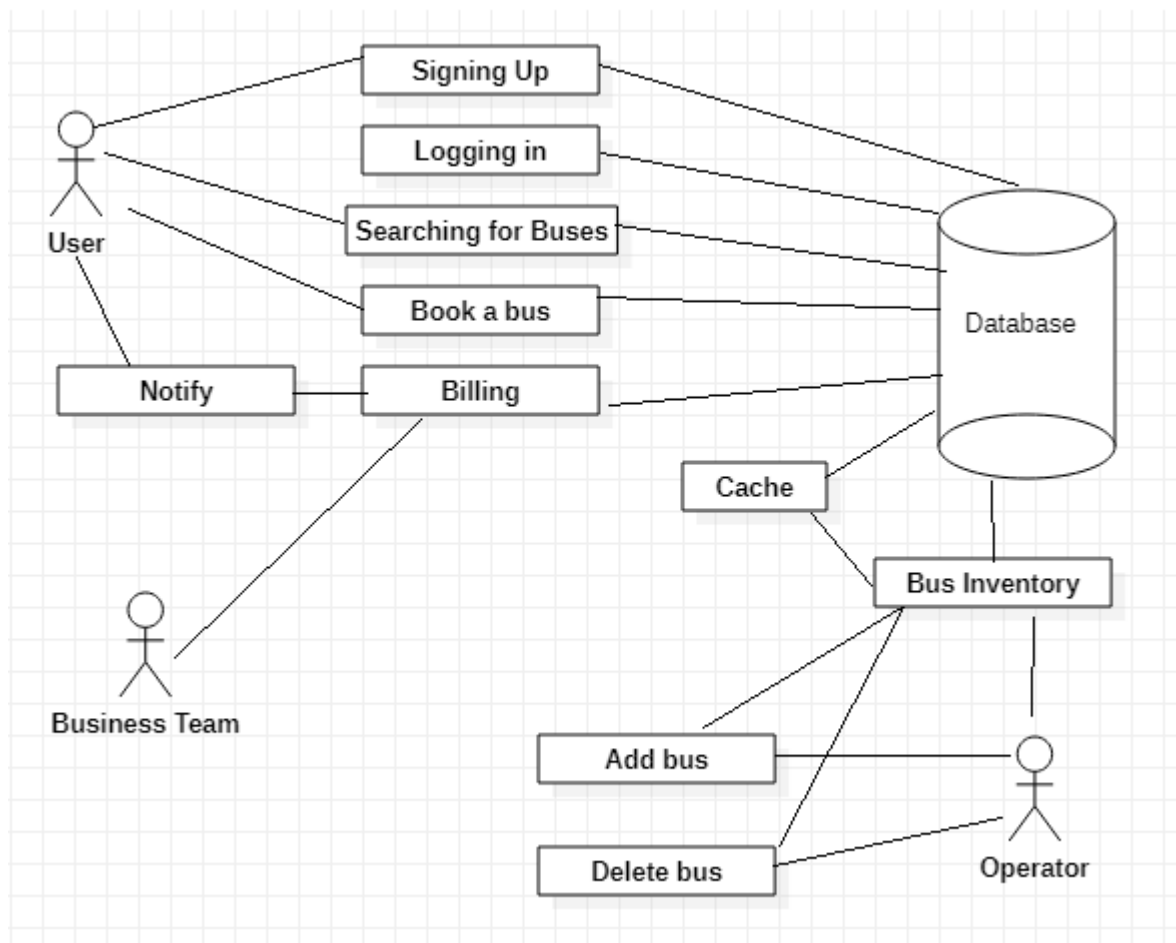


Design Diagrams

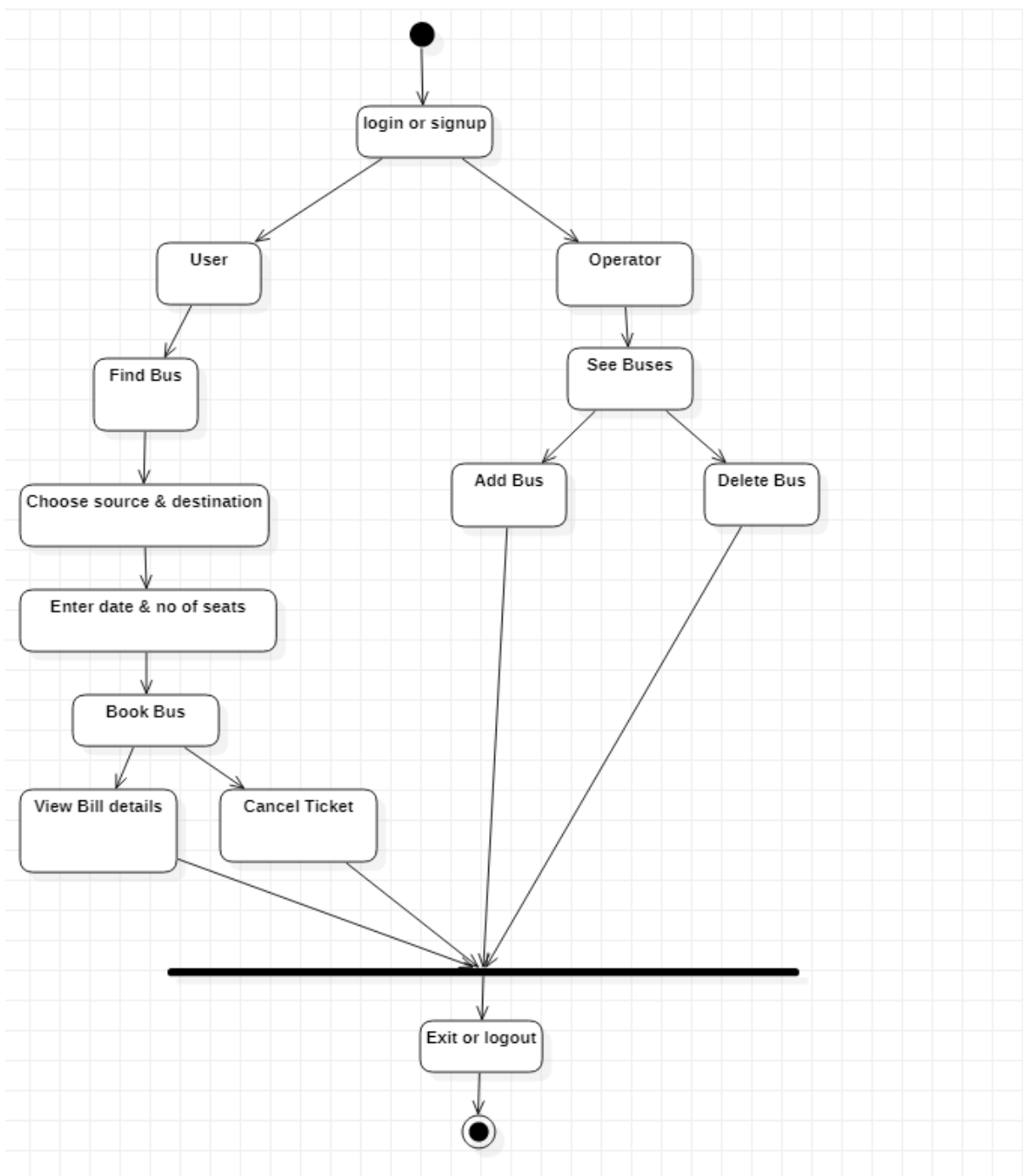
Activity Diagram



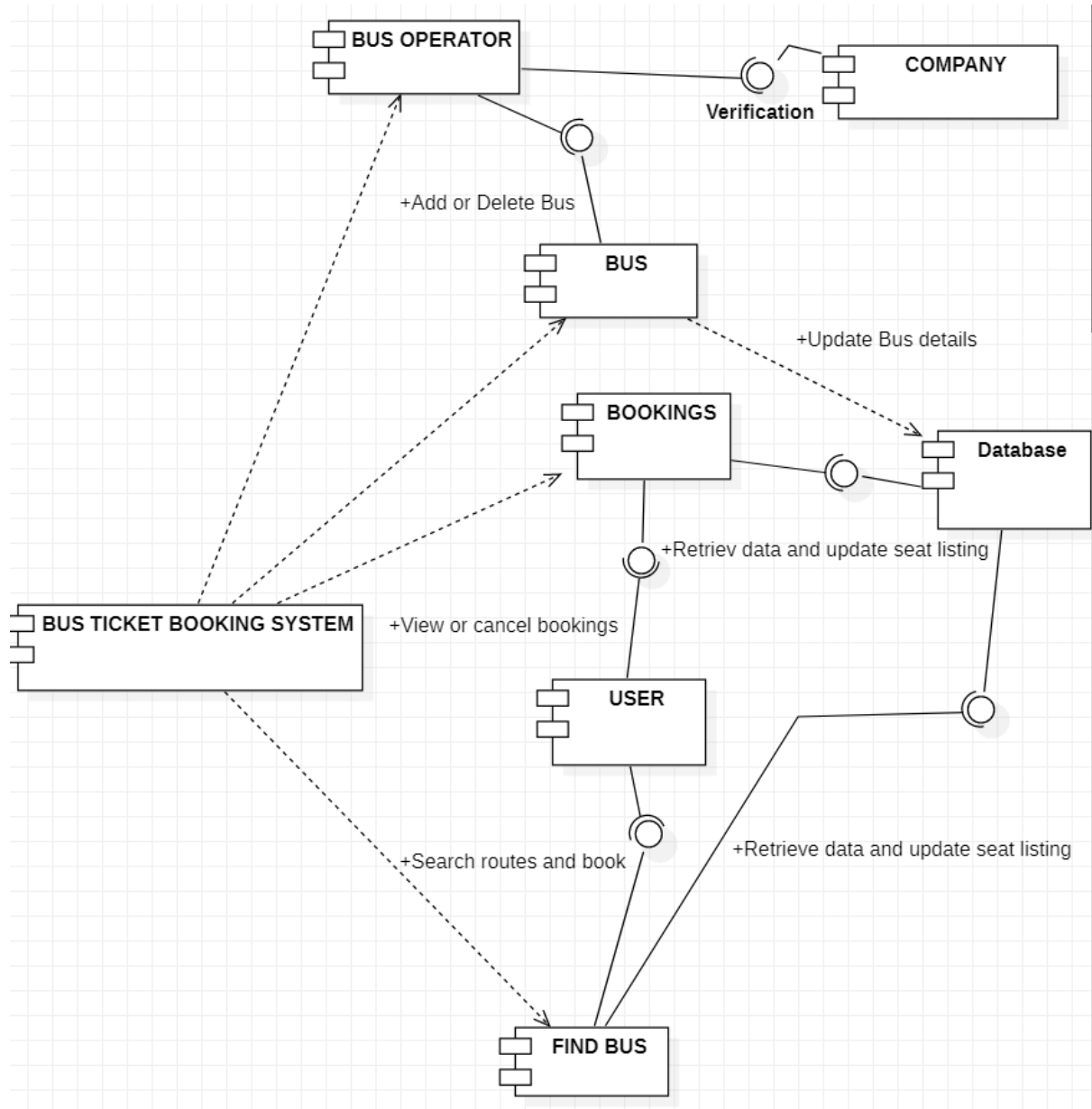
System Architecture Diagram



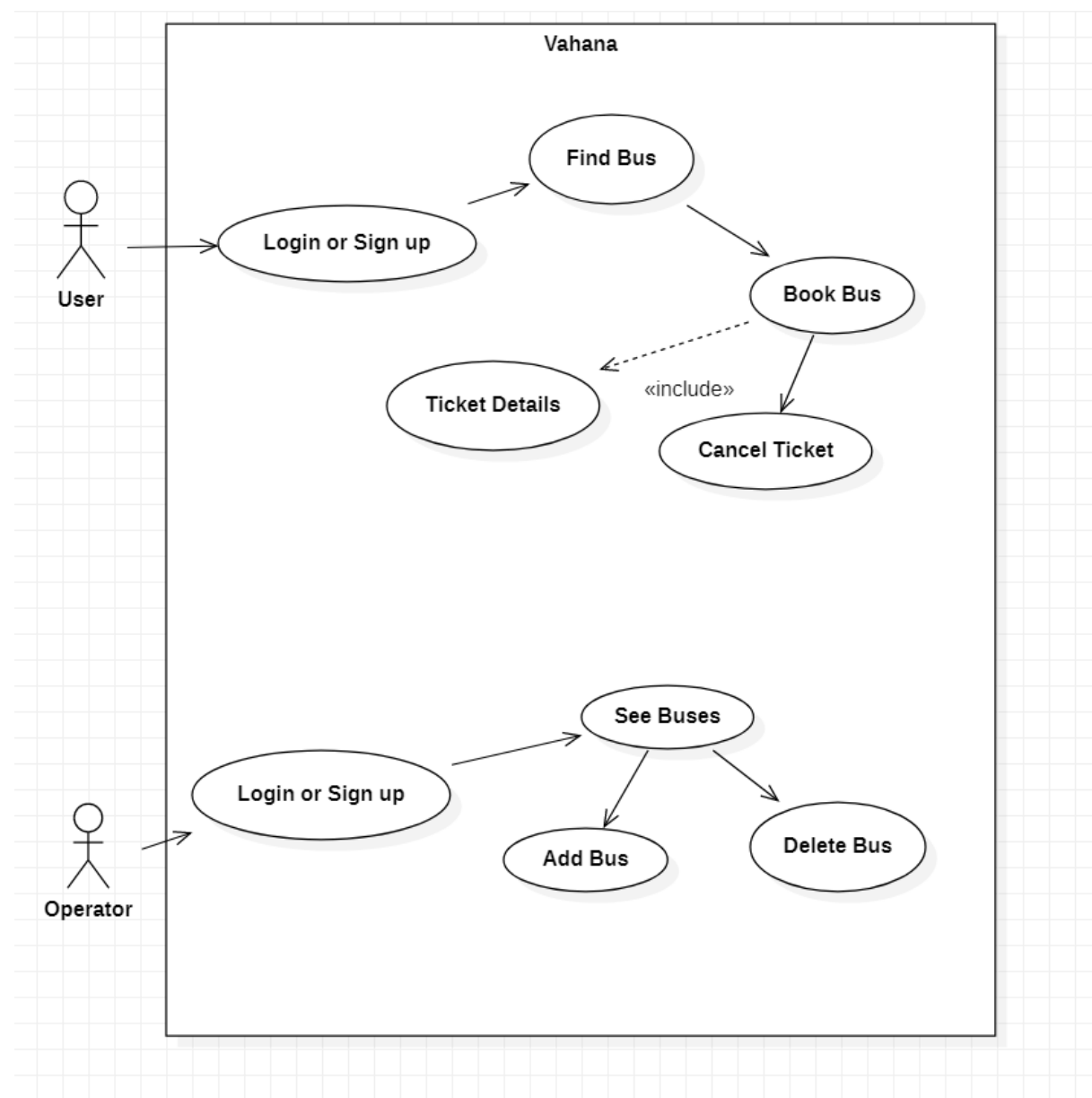
State Diagram



Component Diagram

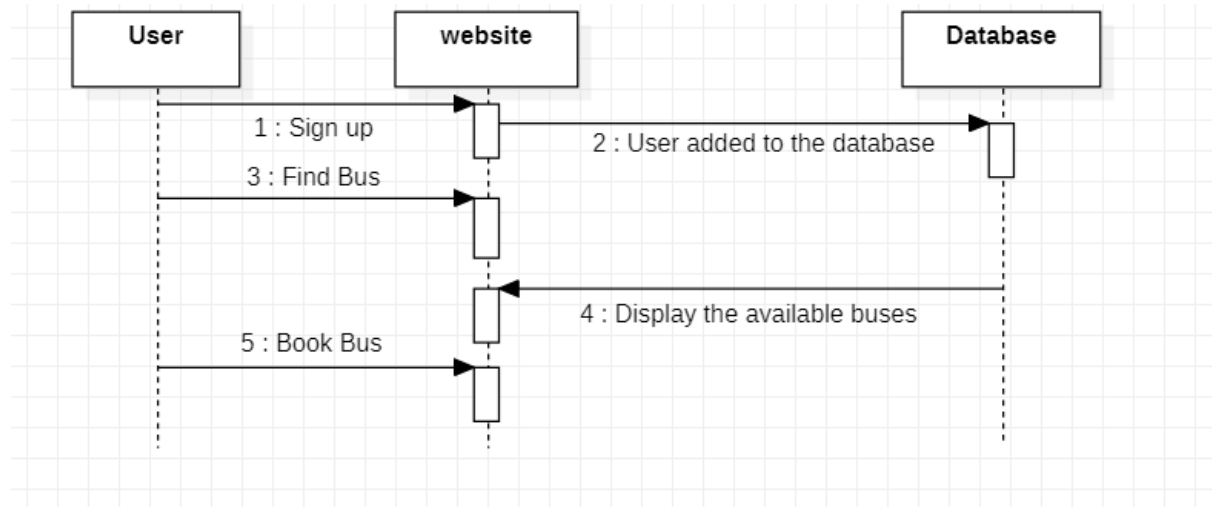


Use Case Diagram

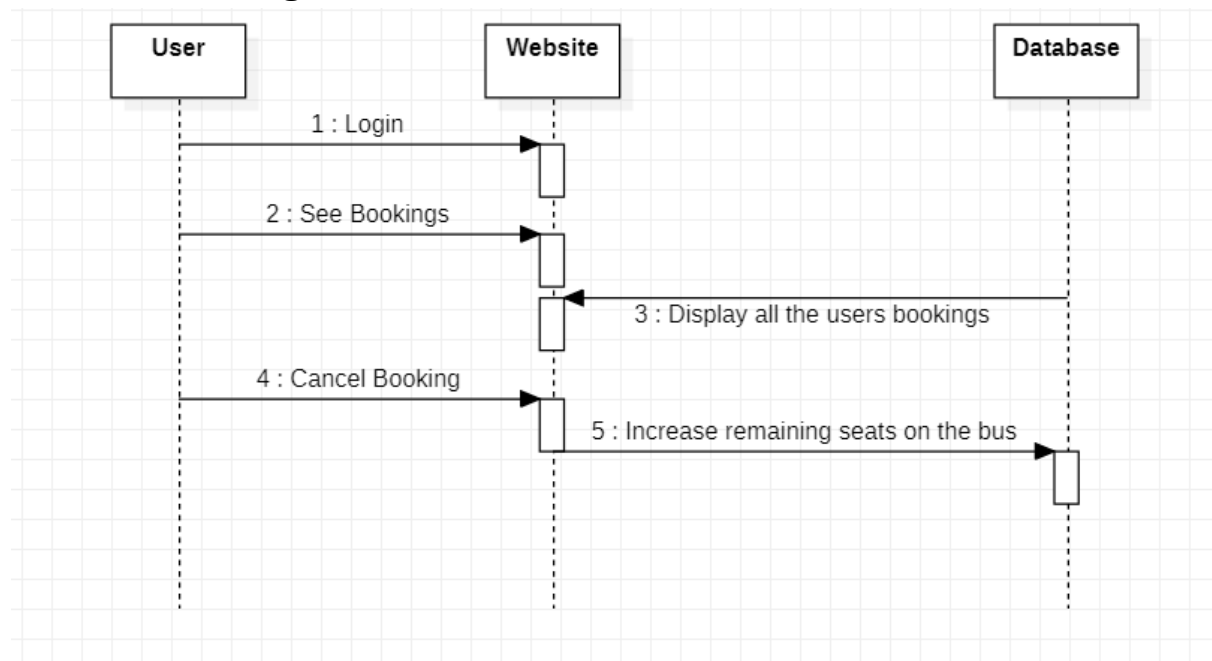


Sequence Diagrams

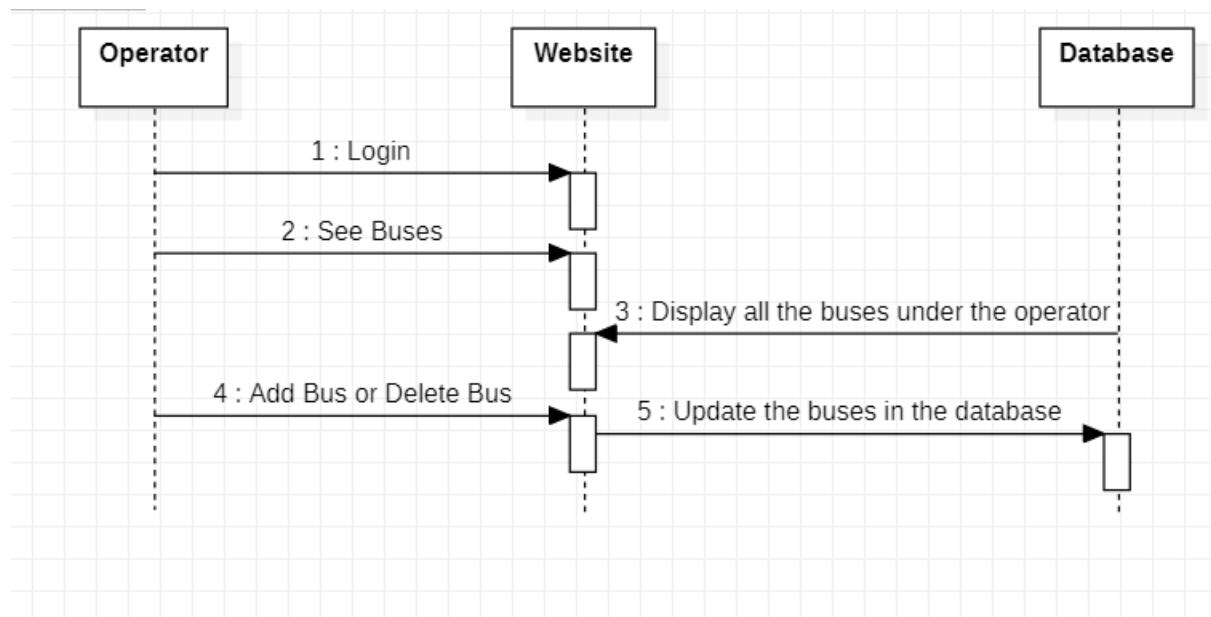
Booking a ticket



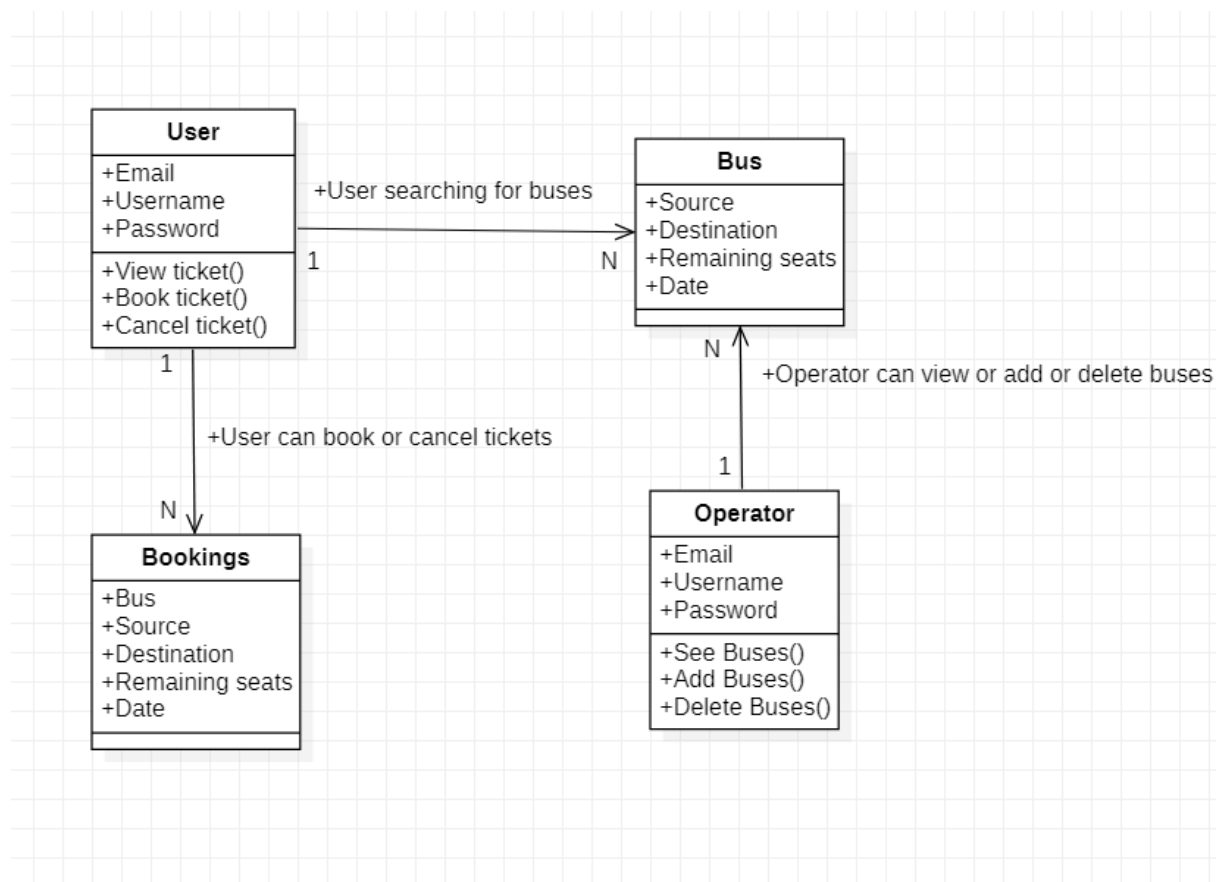
Cancel booking



Operator



Class Diagram



Module Description

User

This module is for authenticating users of the website. The default User model in Django uses a username to uniquely identify a user during authentication. The user can be a customer or an operator. There is no separate module for normal users. If they are a customer, they can find bus routes, book buses, cancel buses and view bookings.

Operator

This module is for authenticating users of the website. While this model uses the default Django user model too, there is also a separate model for operators. The User model is used for authenticating, while the Operator model differentiates the normal users from operators. The operator module has the ability to add and delete bus objects.

Bus

This module is used to store bus objects with their details. This module is pivotal to the entire website, since the project is based on this.

Booking

This module is used to let users book buses. This module gives a list of buses based on the searched requirements. It uses the bus module to create the list.

Anonymous User

The anonymous user can search for and view bus routes but to use any other functionality, they'd have to login.

Test Cases

Test Case ID	Name of Module	Test case description	Pre-conditions	Test Steps	Test data	Expected Results	Actual Result	Test Result
--------------	----------------	-----------------------	----------------	------------	-----------	------------------	---------------	-------------

RL-01	User registration	To test the sign up functionality as user	access the chrome browser	1. Navigate to the website 2. Click on signup 3. Enter details 4. Sign up as User 5. submit	Email: teja@gmail.com Username : teja Password : teja123	Sign up should be successful	Sign up Successful	Pass
RL-11	User Login	To test the login functionality as user	access the chrome browser	1. Navigate to the website 2. Click on login 3. Enter details 4. submit	Username : teja Password : teja123	Login should be successful	Login successful	Pass
RL-21	Operator registration	To test the sign up functionality as operator	access the chrome browser	1. Navigate to the website 2. Click on signup 3. Enter details 4. Sign up as operator 5. submit	Email: Babu@gmail.com Username: Babu Password: boom123123	Sign up should be successful	Sign up Successful	Pass
RL-31	Operator Login	To test the login functionality as operator	access the chrome browser	1. Navigate to the website 2. Click on login 3. Enter details 4. submit	Username: Babu Password: boom123123	Login should be successful	Login successful	Pass
RL-02	User registration	To test the sign up functionality as User	access the chrome browser	1. Navigate to the website 2. Click on signup 3. Enter details 4. Sign up as User 5. submit	Email: tejap@gmail.com Username : teja Password : tejap123	Sign up should be unsuccessful (Same username)	Sign up should be unsuccessful (Same username)	Pass
RL-22	Operator registration	To test the sign up functionality as operator	access the chrome browser	1. Navigate to the website 2. Click on signup 3. Enter details 4. Sign up as operator 5.. submit	Email: Babu@gmail.com Username: Babu Password: boom123123	Sign up should be unsuccessful (Same username)	Sign up should be unsuccessful (Same username)	Pass
RL-12	User Login	To test the login functionality as user	access the chrome browser	1. Navigate to the website 2. Click on login 3. Enter details 4. submit	Username: Babu Password: boom123123	Login should be unsuccessful (User already registered as operator)	Login unsuccessful (User already registered as operator)	Pass

RL-32	Operator Login	To test the login functionality as operator	access the chrome browser	1. Navigate to the website 2. Click on login 3. Enter details 4. submit	Username : teja Password : teja123	Login should be unsuccessful (User already registered as user)	Login unsuccessful (User already registered as user)	Pass
RL-03	User registration	To test the sign up functionality as User	access the chrome browser	1. Navigate to the website 2. Click on signup 3. Enter details 4. Sign up as User 5. submit	Email: Gouthamswa123 Username : Goutham Password : Goutham123	Sign up should be unsuccessful (Credentials are in the wrong format)	Sign up unsuccessful (Credentials are in the wrong format)	Pass
RL-23	Operator registration	To test the sign up functionality as Operator	access the chrome browser	1. Navigate to the website 2. Click on signup 3. Enter details 4. Sign up as Operator 5. submit	Email: Gouthamswa123 Username : Goutham Password : Goutham123	Sign up should be unsuccessful (Credentials are in the wrong format)	Sign up unsuccessful (Credentials are in the wrong format)	Pass

Test Case ID	Name of Module	Test case description	Pre-conditions	Test Steps	Test data	Expected Results	Actual Result	Test Result
US-01	User searches for buses	User enters source location and not the destination and date	User should have a working browser but does not have to be logged in	1. Navigate to the "Find Bus" tab 2. Enter Source location 3. Click on "Find bus"	Source: Bangalore	List of buses with Bangalore as the source location	List of buses with Bangalore as the source location	Pass
US-02	User searches for buses	User enters any combination of source, destination and date	User should have a working browser but does not have to be logged in	1. Navigate to the "Find Bus" tab 2. Enter travel details 3. Click on "Find bus"	Source: Mangalore, Destination: Bangalore Date: 16-04-2021	List of buses with Mangalore as the source, Bangalore as the destination available on April 16th 2021	List of buses with Mangalore as the source, Bangalore as the destination available on April 16th 2021	Pass
US-03	User searches for buses	User enters any combination of source, destination and date. The entered source and/or destination	User should have a working browser but does not have	1. Navigate to the "Find Bus" tab	Source: Mangalore, Destination: Mumbai Date:	Display error message "Sorry no buses available"	Error Message displayed: "Sorry no buses available"	Pass

		with/without giving a date has no bus plying.	to be logged in	2.Enter travel details 3. Click on “Find bus”	16-04-2021			
US-11	User attempts to book a bus	User searches for bus routes to book and attempts to book without being logged in	User is not logged in	1. Navigate to the “Find Bus” tab 2.Enter travel details 3. Click on find bus 4. Enter the bus ID of the route to take 5. Enter number of seats 6.Click on “Book Bus”	Bus ID: 8 Number of seats: 5	Redirect to login page. (the user can book only when logged in) The remaining seats for the bus with ID:8 should remain unchanged in the database	Redirect to login page. Remaining seats unchanged for Bus with ID:8 in the database	Pass
US-12	User attempts to book a bus	User searches for bus routes to book and attempts to book when logged in	User is logged in	1. Navigate to the we 2.Enter travel details 3. Click on find bus 4. Enter the bus ID of the route to take 5. Enter number of seats 6.Click on “Book Bus”	Bus ID: 8 Number of seats: 5	Booking should happen, and the user should be redirected to the page with a confirmation message and an option to view Bill Details. The remaining seats for the bus with ID:8 changes by 5 seats.	User is redirected to the page with a confirmation message and an option to view Bill Details. The remaining seats for the bus with ID:8 changes by 5 seats.	Pass
US-13	User attempts to book a bus	User searches for bus routes to book and attempts to book more seats than available for a listed bus	User is logged in	1. Navigate to the we 2.Enter travel details 3. Click on find bus 4. Enter the bus ID of the	Bus ID: 8 Number of seats: 55	Display error message “Sorry select fewer number of seats”. There should be no change to the number of	Error Message displayed : “Sorry select fewer number of seats”. Number of seats remains unchanged in the database.	Pass

				route to take 5. Enter more seats than the listed available seats for a bus given on the "List of Buses" table. 6. Click on "Book Bus"		seats in the database.		
US-21	See Bookings	User attempts to view previously booked tickets without logging in	User is not logged in	1. Navigate to the "See Bookings" tab	NA	Redirect to Login page	Redirect to Login page	Pass
US-22	See Bookings	User attempts to view previously booked tickets being logged in	User is logged in	1. Navigate to the "See Bookings" tab	NA	List of Buses previously booked by the user (includes canceled reservations)	List of Buses previously booked by the user (includes canceled reservations)	Pass
US-31	Cancel Bookings	User attempts to cancel previously booked tickets.	User is logged in.	1. Navigate to the "See Bookings" tab. 2. Enter Booking ID to cancel. 3. Click on "Cancel Bus".	Booking ID: 3	Booking with ID:3 should get canceled for user vinay. The remaining seats should be updated in the database for the bus canceled by vinay. The change from "BOOKED" to "CANCELED" should be reflected on the "List of Buses" table.	Booking with ID:3 is canceled for user vinay. The remaining seats are updated in the database for the bus canceled by vinay. The change from "BOOKED" to "CANCELED" is reflected on the "List of Buses" table.	Pass

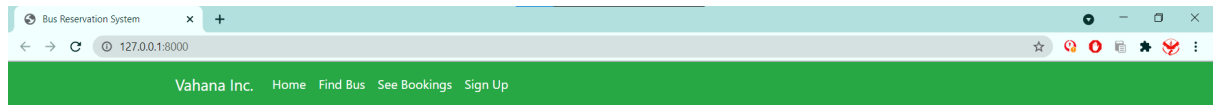
Test Case ID	Name of Module	Test case description	Pre-conditions	Test Steps	Test data	Expected Results	Actual Result	Test Result
--------------	----------------	-----------------------	----------------	------------	-----------	------------------	---------------	-------------

OP-01	Operator viewing buses under it	To check the list of buses that are under the operator	There are already buses under the operator, and the operator has to be signed in	1.Sign in as an operator. 2.Click on “Check Buses”.	Username: Babu Password: boom123123	List of buses under the operator “Babu”	List of buses under the operator “Babu”	Pass
OP-11	Operator adding a bus under it	To add a new bus that plies under that operator	The operator has to be signed in	1.Sign in as an operator. 2.Click on “Check Buses”. 3.Click on “Add Bus”	Username: Babu Password: boom123123	A new bus called “AC Bus X” should be added	“AC Bus X” got added	Pass
OP-21	Operator deleting bus under it	To delete a bus that plies under that operator	The operator has to be signed in	1.Sign in as an operator. 2.Click on “Check Buses”. 3.Click on the “Delete Bus” button corresponding to the bus to be deleted.	Username: Babu Password: boom123123	The bus called “AC Bus X” should be deleted	“AC Bus X” got deleted	Pass
OP-02	Operator viewing buses under it	To check the list of buses that are under the operator	There are already buses under the operator, and the operator has to be signed in	1.Sign in as an operator. 2.Click on “Check Buses”	Username: Ash Password: boom123123	List of buses under the operator “Ash”	List of buses under the operator “Ash”	Pass
OP-12	Operator adding a bus under it	To add a new bus that plies under that operator	The operator has to be signed in	1.Sign in as an operator. 2.Click on “Check Buses”. 3.Click on “Add Bus”	Username: Ash Password: boom123123	A new bus called “Quick Bus” should be added	“Quick Bus” got added	Pass
OP-22	Operator deleting bus under it	To delete a bus that plies under that operator	The operator has to be signed in	1.Sign in as an operator. 2.Click on “Check Buses”. 3.Click on the “Delete Bus”	Username: Ash Password: boom123123	The bus called “Quick Bus” should be deleted	“Quick Bus” got deleted	Pass

				button corresponding to the bus to be deleted.				
OP-03	Operator viewing buses under it	To check the list of buses that are under the operator	There are already buses under the operator, and the operator has to be signed in	1.Sign in as an operator. 2.Click on “Check Buses”	Username: Fred Password: boom123123	List of buses under the operator “Fred”	List of buses under the operator “Fred”	Pass
OP-13	Operator adding a bus under it	To add a new bus that plies under that operator	The operator has to be signed in	1.Sign in as an operator. 2.Click on “Check Buses”. 3.Click on “Add Bus”	Username: Fred Password: boom123123	A new bus called “Bus 1” should be added	“Bus 1” got added	Pass
OP-23	Operator deleting bus under it	To delete a bus that plies under that operator	The operator has to be signed in	1.Sign in as an operator. 2.Click on “Check Buses”. 3.Click on the “Delete Bus” button corresponding to the bus to be deleted.	Username: Fred Password: boom123123	The bus called “Bus X” should be deleted	“Bus X” got deleted	Pass

Final Outcome

Home Page



Log in

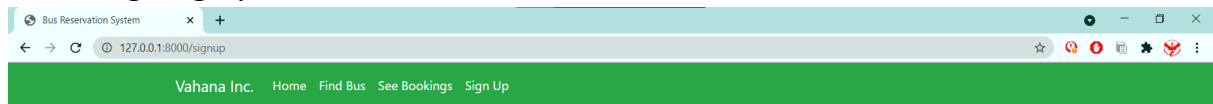
Username:

Password:

Are you an operator? ☐

[Sign in](#)

Signing up



Sign up!

Email:

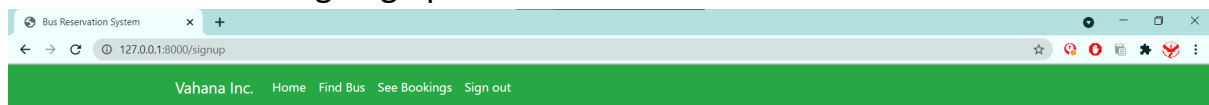
Username:

Password:

Are you an operator? ☐

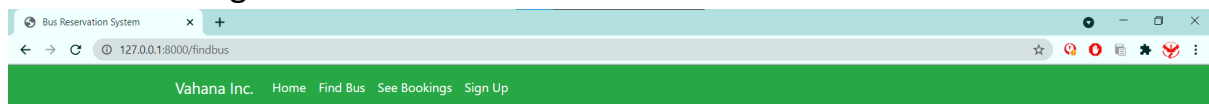
[Sign up](#)

Screen after signing up



Thank you for signing up!

Searching for a bus



Find bus

From	<input type="text" value="Bangalore"/>
To	<input type="text" value="Chennai"/>
Date	<input type="text" value="dd-mm-yyyy"/>

Viewing available buses(You don't need to login for this)

8

Vahana Inc. Home Find Bus See Bookings Sign Up

List of buses

ID	NAME	SOURCE	DESTINATION	NUM OF SEATS	REM SEATS	PRICE	DATE	TIME
3	bus x	Bangalore	Chennai	60	60	500.00	April 29, 2021	6 a.m.

Choose bus to book

Bus ID

Number of seats

Book bus

Booking seats for a bus

Vahana Inc. Home Find Bus See Bookings Sign out

Booking Confirmation

Your booking has been confirmed!

Thank you!

Please keep the bill details. This is your ticket.

Bill details

OK

Confirmation Bill (Ticket)

The screenshot shows a web browser window with the address bar displaying "127.0.0.1:8000/bookings". The page has a green header with "Vahana Inc." and navigation links: "Home", "Find Bus", "See Bookings", and "Sign out". The main content area displays a "Booking Confirmation" message: "Your booking has been confirmed! Thank you! Please keep the bill details. This is your ticket." Below this message is a green button labeled "Bill details". A modal window titled "Ticket Details" is open, showing the following information:

- Bus name: bus x
- Starting point: Bangalore
- Destination point: Chennai
- Number of seats: 5
- Price: 500.00
- Cost: 2500.00
- Date: April 29, 2021
- Time: 6 a.m.

The modal has a "Close" button at the bottom right. An "OK" button is visible on the right side of the main page.

List of buses the user has booked

The screenshot shows a web browser window with the address bar displaying "127.0.0.1:8000/seebookings". The page has a green header with "Vahana Inc." and navigation links: "Home", "Find Bus", "See Bookings", and "Sign out". The main content area displays a "List of buses" table with the following data:

BOOKING ID	USER NAME	BUS NAME	SOURCE	DESTINATION	NUM OF SEATS	PRICE	DATE	TIME	STATUS
5	goutham	bus x	Bangalore	Chennai	5	500.00	April 29, 2021	6 a.m.	BOOKED
6	goutham	bus x	Bangalore	Chennai	5	500.00	April 29, 2021	6 a.m.	BOOKED
7	goutham	bus x	Bangalore	Chennai	5	500.00	April 29, 2021	6 a.m.	BOOKED
8	goutham	bus x	Bangalore	Chennai	0	500.00	April 29, 2021	6 a.m.	CANCELLED

Below the table is a section titled "Choose bus to book" with a "Bus ID" label and a text input field containing the value "7". A red button labeled "Cancel bus" is located below the input field.

After cancelling a booking

Bus Reservation System

127.0.0.1:8000/seebookings

Vahana Inc. Home Find Bus See Bookings Sign out

List of buses

BOOKING ID	USER NAME	BUS NAME	SOURCE	DESTINATION	NUM OF SEATS	PRICE	DATE	TIME	STATUS
5	goutham	bus x	Bangalore	Chennai	5	500.00	April 29, 2021	6 a.m.	BOOKED
6	goutham	bus x	Bangalore	Chennai	5	500.00	April 29, 2021	6 a.m.	BOOKED
7	goutham	bus x	Bangalore	Chennai	0	500.00	April 29, 2021	6 a.m.	CANCELLED
8	goutham	bus x	Bangalore	Chennai	0	500.00	April 29, 2021	6 a.m.	CANCELLED

Choose bus to book

Bus ID

Cancel bus

Updated seats for the bus that got booked

Bus Reservation System

127.0.0.1:8000/findbus

Vahana Inc. Home Find Bus See Bookings Sign out

List of buses

ID	NAME	SOURCE	DESTINATION	NUM OF SEATS	REM SEATS	PRICE	DATE	TIME
3	bus x	Bangalore	Chennai	60	50	500.00	April 29, 2021	6 a.m.

Choose bus to book

Bus ID

Number of seats

Book bus

Signing up as an operator

Bus Reservation System

127.0.0.1:8000/signup

Vahana Inc. Home Find Bus See Bookings Sign Up

Sign up!

Email:

Username:

Password:

Are you an operator? ☒

[Sign up](#)

List of buses under operator(this operator is newly registered)

Operator View

127.0.0.1:8000/seebuses

Vahana Inc. Home Check Buses Sign out

List of buses

ID	NAME	SOURCE	DESTINATION	NUM OF SEATS	REM SEATS	PRICE	DATE	TIME
Add Bus								

Adding a bus to operator

Operator View

127.0.0.1:8000/addbus

Vahana Inc. Home Check Buses Sign out

Add a Bus

Bus name:

MAZDAspeed

Source:

Mysore

Dest:

Mangalore

Nos:

60

Rem:

60

Price:

100

Date:

2021-04-28

Time:

06:00:00

Name:

operator@gmail.com

Submit

List of buses under operator (after adding a bus)

Operator View

Vahana Inc. Home Check Buses Sign out

List of buses

ID	NAME	SOURCE	DESTINATION	NUM OF SEATS	REM SEATS	PRICE	DATE	TIME
11	MAZDAspeed	Mysore	Mangalore	60	60	100.00	April 28, 2021	6 a.m.

Add Bus Delete Bus

Trying to sign in as an operator (when you are a general user)

Bus Reservation System

Vahana Inc. Home Find Bus See Bookings Sign Up

Welcome to Vahana Bus Services

Log in

Username: goutham

Password:

Are you an operator? ☒

You have been logged out

Sign in

Welcome to Vahana Bus Services

Log in

Username:

Password:

Are you
an
operator? ☐

You do not have an account registered as an operator.

[Sign in](#)