

BOSTON UNIVERSITY
School of Arts & Sciences
Department of Computer Science
Course Title: Software Engineering | Course #: CS411. | Section: B
Professor Ahmed Ibrahim

Car Rental System
Project Report

Group Eight:
Li, Haowei U32332746
Wang, Yichen U38122978
Wu, Simo U94567414
Yan, Wenxuan U83370649
Zhang, Runze U11045612

	2
CONTENTS	
Introduction	3
Problem Definition	3
Project Objectives	4
Project Deliverables	4
Stakeholder	5
Success/Acceptance Criteria	6
Use Case Diagram	7
Selected Use Case Descriptions	8
Sequence Diagram	10
Class Diagram	11
State Diagram	12
Subsystem Decomposition	12
ER Diagram	13
GitHub link to the project source code	13
Project Management	14
WBS:	14
Task Assignment Matrix:	15
Gantt Chart:	15
Task Tracker:	16
Test Cases	17
Work Partition	19
Conclusion	20
Reference	20

Introduction

The significance of this project is to provide a car rental system for the car rental companies, which aim to become young user-friendly companies with affordable rates. To ensure the quality of the project, we developed a system that consists of durable, effective architecture that can be used on computers. This system is promised to keep a low maintenance cost for long-term benefits.

The car rental companies can list and manage their cars on the platform and easily collect the information and revenue as needed. With the system, the customers are able to rent and return a car.

In this system, we will have a interface that contains functionalities including:

1. A car section allows adding and deleting the cars based on the lender company decisions, and the system also stores all the attributes of the cars.
2. A reservation section allows customers to rent and return a car. The system also keeps records of the cars rented and the revenue.
3. A customer section allows the registration and deletion of customers. The system will store all the user information (e.g., registration, insurance company, license, address, phone number, payment information).

Problem Definition

Large car rental franchises often have strict and complex rules. Small businesses want to enter the car rental market and provide users with a more effortless car rental experience. However, these small car rental companies generally use paperwork to record customer information, car information, rental orders, etc. Our goal is to help these small rental companies provide them with an electronic car rental system. This more convenient and intuitive system can simplify the car rental and recording processes and reduce manual work.

Project Objectives

- The system's software and business requirements will be specified by Oct 30th.
- The system's car class will be constructed as required by Nov 7th.
- The system's reservation class will be built as required by Nov 14th.
- The system's customer class will be built as required by Nov 21st.
- All the specified classes will perform the first test on Nov 22nd.
- The first revision process will be finished on Nov 23th.
- All the specified classes will perform the second test on Nov 24th.
- The second revision process will be finished on Nov 25th.
- All the specified classes will perform the final test on Nov 26th.
- The final revision process will be finished on Nov 27th.
- The system will be up online on Nov 28th.

Project Deliverables

- List of requirements - Oct 30th
- Use case diagram(s) - Nov 5th
- Use case Descriptions - Nov 5th
- Sequence diagram(s) - Nov 5th
- Class diagram(s) - Nov 5th
- Car Rental System Prototype - Nov 22nd
- Working Program - Nov 28th
- Instruction for the system - Nov 30th

Stakeholder

The stakeholders of the project consist of car rental companies, customers, and platform developers:

- Car Rental Company:
They look forward to the completion of this project to perform business as required. The acceptance of the system influences the project duration and the complexity of the source code.
- Customers:
For the customers, they need a direct and efficient system to rent cars. Their feedback on the system will impact the commercial performance of the car rental company and the system. The size of the users' community and the number of interactions between the users and the system will influence the system's maximum load capacity and efficiency.
- Platform developers:
We are responsible for constructing an executable architecture that allows car rental companies and customers to use their functions. Our efforts put into the system will determine the system's efficiency and life expectancy.

Success/Acceptance Criteria

- Car Rental Company
 - Companies can add cars, delete cars, and change car rates.
 - The platform can record transactions between customers and the company.
 - Companies can collect the order details for annual financial reports using the system.
 - A stable database maintains all information, including the car information, customer information, revenue information, etc.
- Customers
 - Customers can register accounts and make reservations.
 - Customers can finish the rental car process with the system.
 - Customers can finish the return car process with the system.
- Platform Developers
 - We constructed an executable architecture.
 - We can perform constant optimization as well as 24/7 customer support (i.e., debugging)
 - We want to provide a well-constructed program to receive pays and positive feedback.

Use Case Diagram

Our use case diagram is designed to provide services to the car rental company, including allowing the company to manage the car list, manage the customer list, record the order and revenue, and enable their customers to register through the program and finish the whole car rental process.

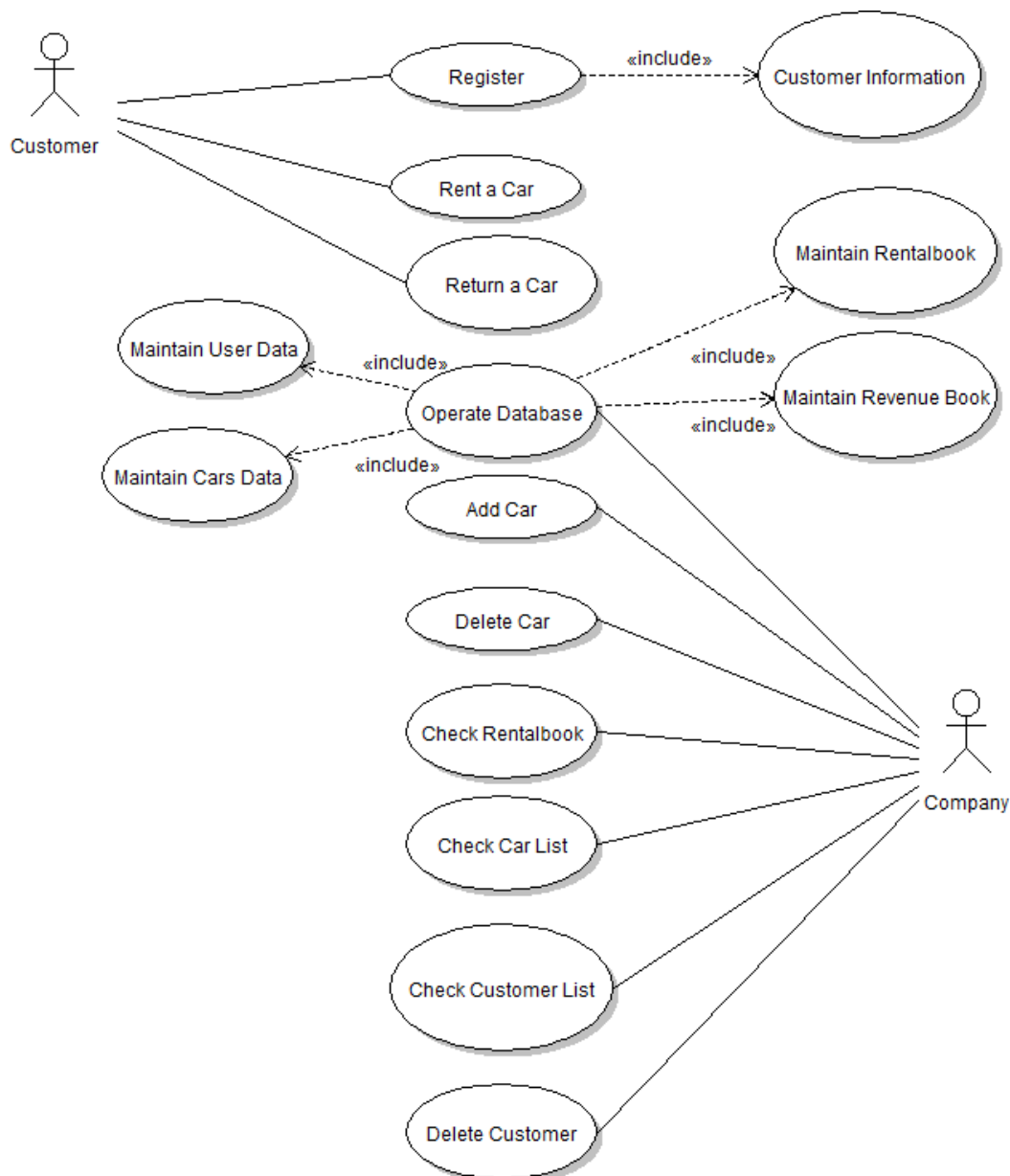


Figure 7.1 Use Case Diagram

Selected Use Case Descriptions

Use Case ID:	A
Use Case Name:	Add a Car
Relevant requirements:	list of cars, a car type, a car's brand model, car condition, car daily rate
Primary Actor:	Owner
Pre-conditions:	The system must have a car data database. The car must exist.
Post-conditions:	The car data database should have the required information. The users can search the car by ID, name, type, color, and level. The order system should be able to add the car. The revenue system should be able to record the income for the car.
Basic Flow or Main Scenario:	<ol style="list-style-type: none"> 1. The owner has access to the system. 2. The owner type in the car information, which include name, ID, type, color, level, price, insurance, status, and condition 3. The car list is updated with the added car.
Extensions or Alternate Flows:	Expand the car list if it is full, then add the car.
Exceptions:	<p>If the car ID exists, ask for a new car ID</p> <p>If the car daily rate is higher than the same type of cars, reject the demand.</p>
Related Use Cases:	Maintain Car Data

Use Case ID:	B
Use Case Name:	Rent a Car
Relevant requirements:	list of cars, car's availability, current order information
Primary Actor:	Customer
Pre-conditions:	The customer must exist. There must be at least one available car in the car database.
Post-conditions:	The record is inside the current order system.
Basic Flow or Main Scenario:	<ol style="list-style-type: none"> 1. The customer searches for the car they want. 2. If the car is currently rented, the customer keeps on searching until they find the car they want is available. 3. The customer chooses the car. 4. The system updates the car as rented.
Extensions or Alternate Flows:	The customer will rent more than one car. The customer will not come to get the car.
Exceptions:	If the car is currently rented, throw an exception, and the user cannot rent the car. If there is not such a car, throw an exception, and the user cannot rent the car. If I am too handsome, I rent a car for free, for life; no return is required.
Related Use Cases:	Maintain Car Data Maintain Reservation Data

Sequence Diagram

In the sequence diagram, the use case "Add a Car" will first need the company to access the system, and the system will provide what attributes it needs to have. Then, after the company enters the required attributes, the system will update the Car list and show it to the company. For the use case Rent a Car, the customer will first need to enter the User ID. After the system verifies the User ID, the customer needs to enter the car name that he wants to rent. Then, the system will mark the car as rented and record the order.

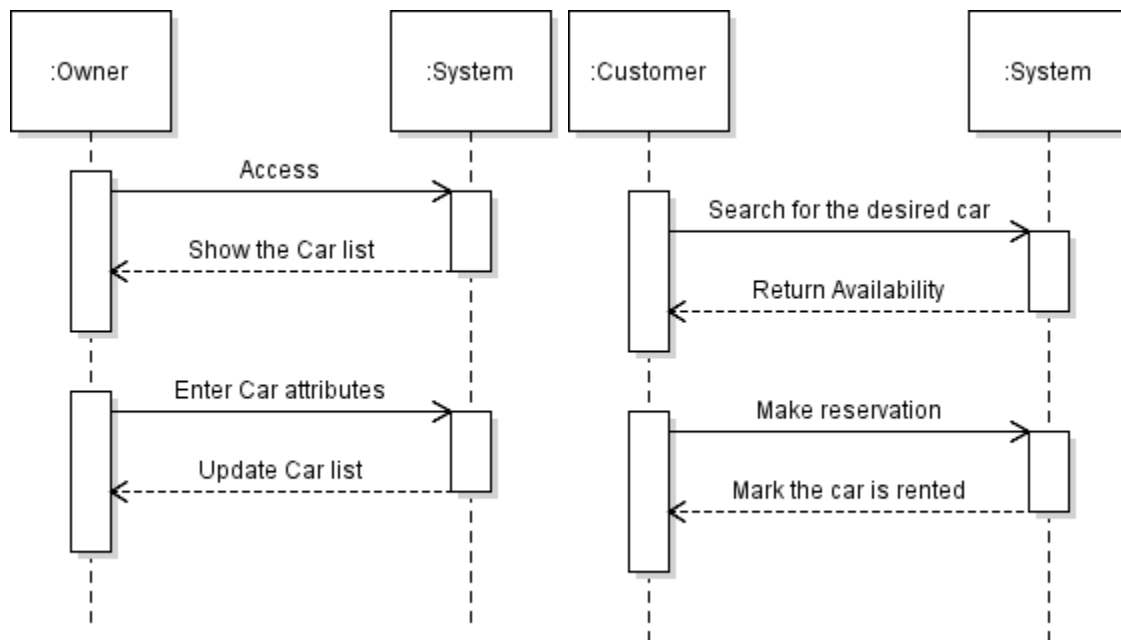


Figure 10.1 Sequence Diagram

Class Diagram

In the class diagram, cars and customer information could exist independently, and the car rental system is constituted by the order and revenue records. As shown in the diagram, we have corresponding attributes and methods in different classes.

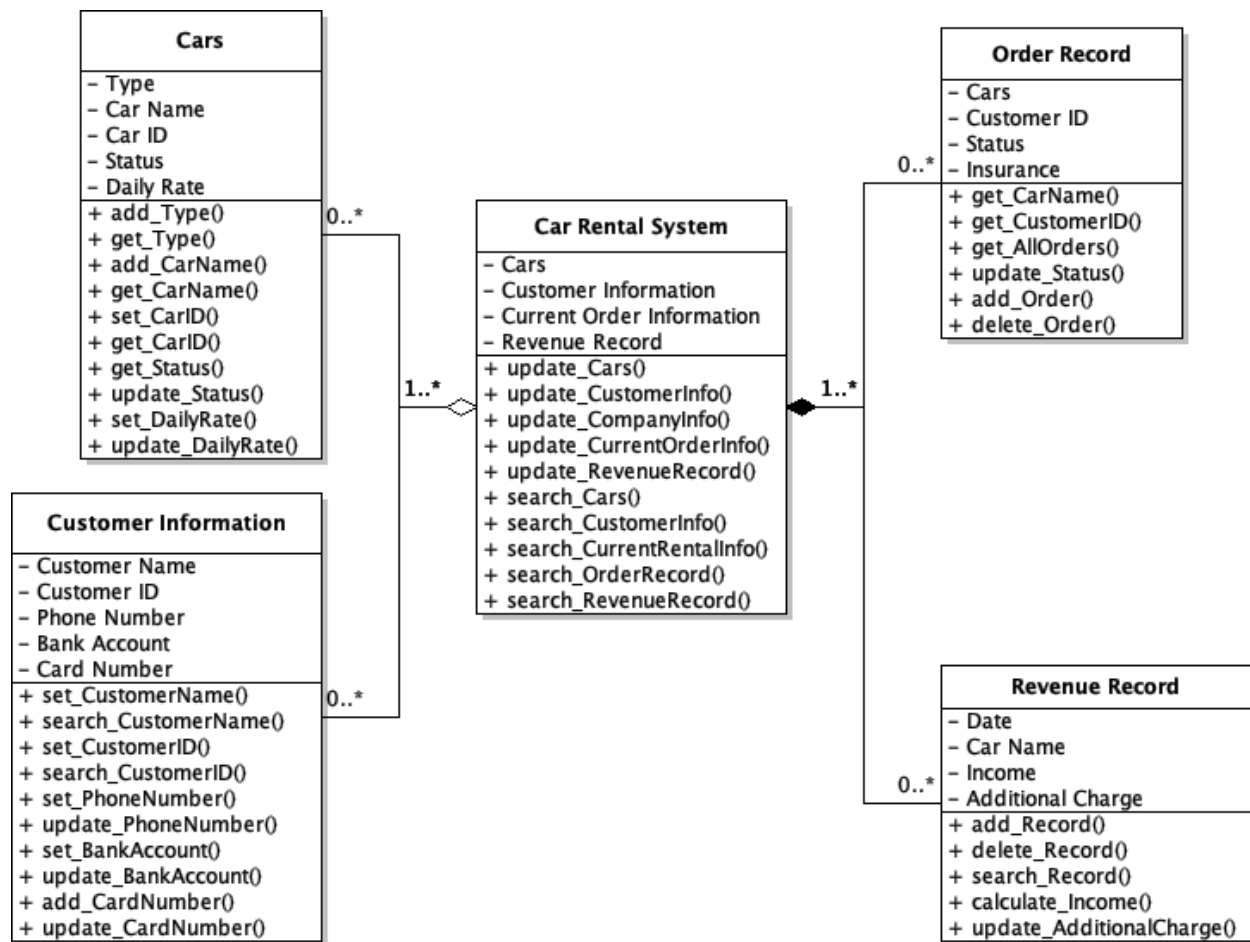


Figure 11.1 Class Diagram

State Diagram

The State Diagram could intuitively and clearly show the process of the car rental system. Customers can enter their Name, ID, and bank information to register. After they have a valid user ID, they can rent a car by entering their User ID and the car's name. If the car is unavailable, the customer needs to enter another car's name. Then, they can return the car by using similar operations plus the return date.

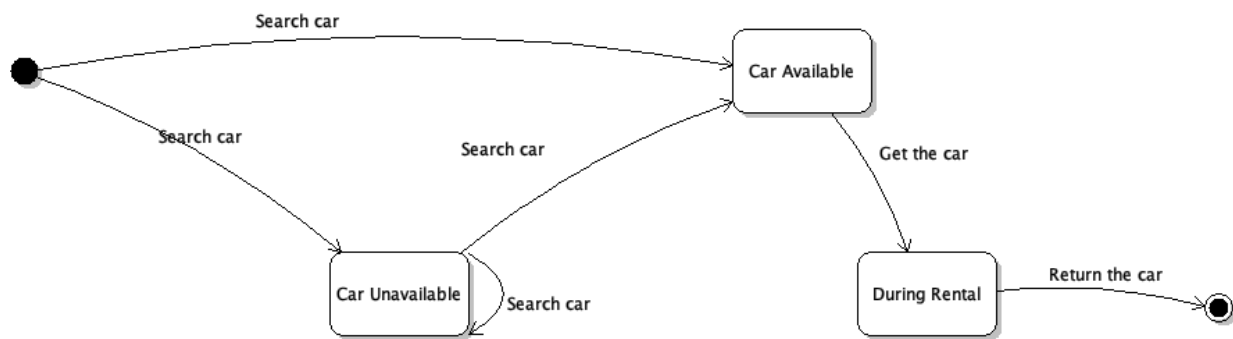


Figure 11.2 State Diagram

Subsystem Decomposition

Our car rental system is composed of four subsystems: customer system, car system, order system, and revenue system.

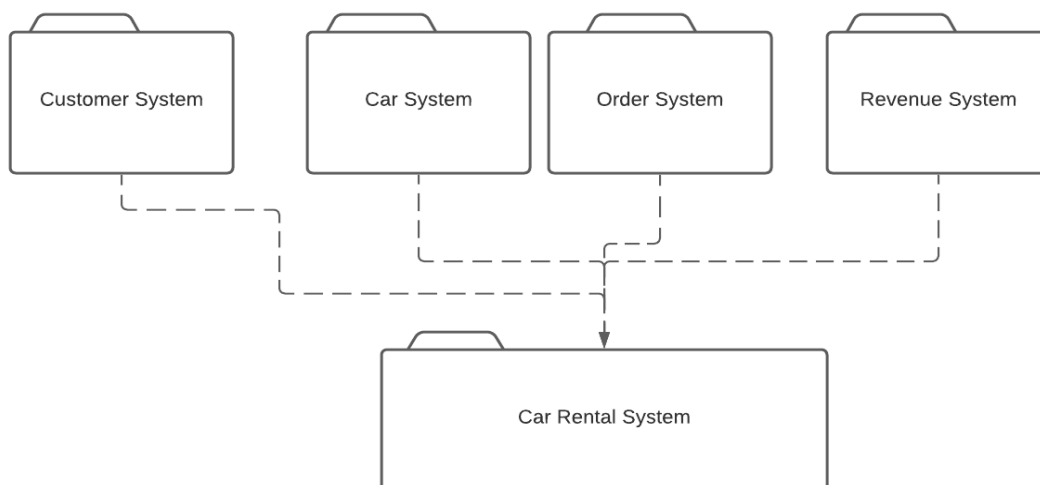


Figure 12.1 Subsystem Decomposition

ER Diagram

Our ER diagram has five entities: customer, car, main system, order system, and revenue system.

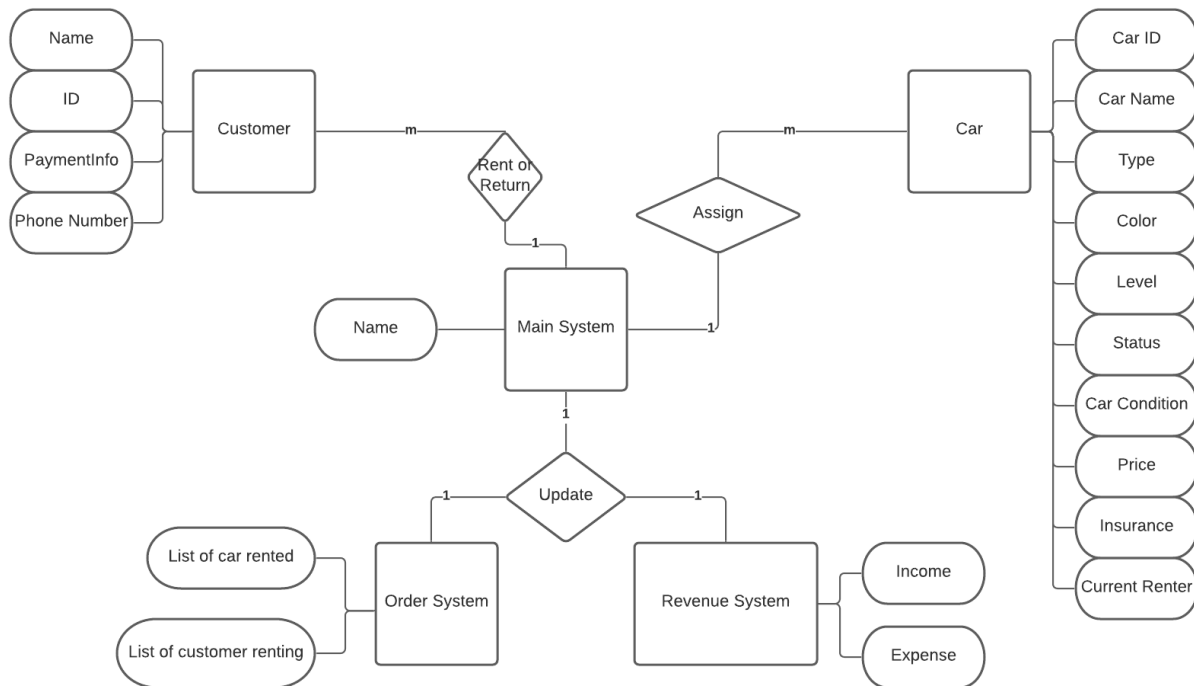


Figure 13.1 ER Diagram

GitHub link to the project source code

<https://github.com/runzeeee/CS411CarRentalProject.git>

Project Management

WBS:

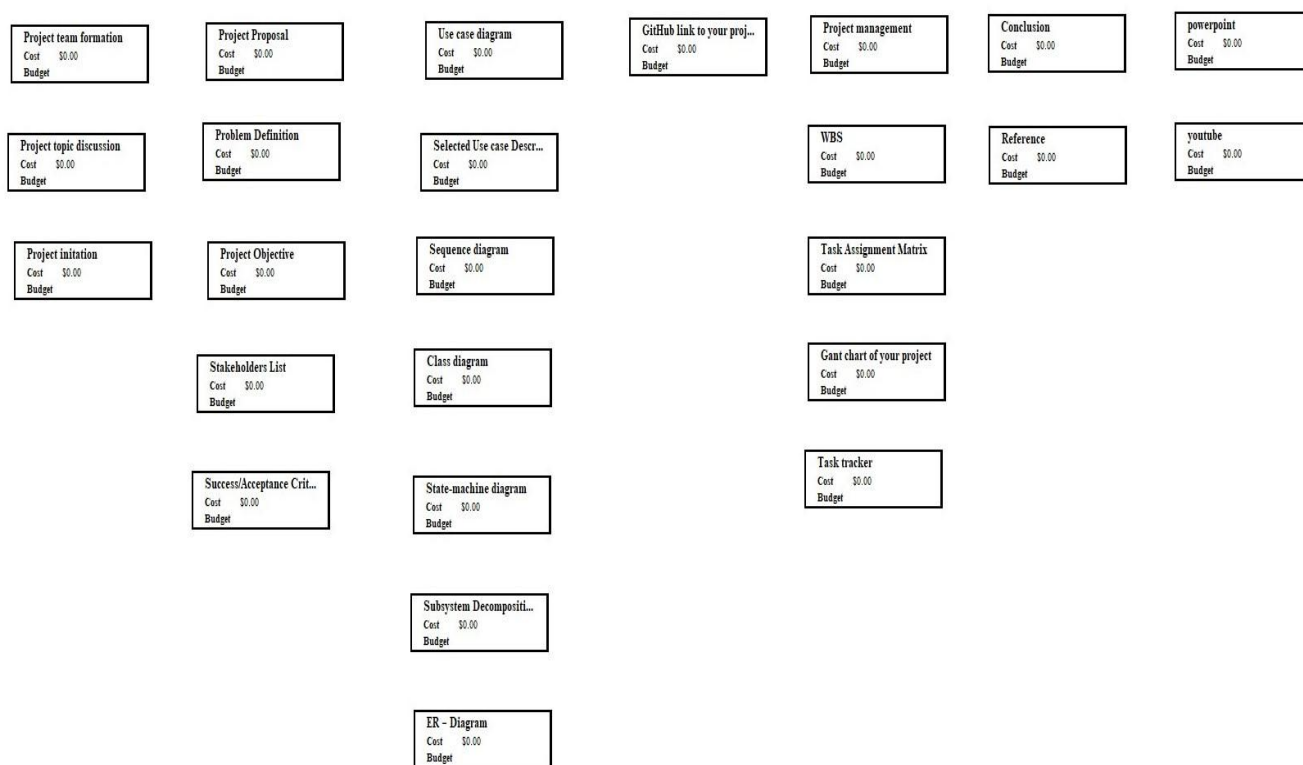


Figure 14.1 WBS Diagram

Task Assignment Matrix:

		Name	Duration	Start	Finish	Predecessors	Resource Names
1	✓	Project team formation	5 days	9/21/21 8:00 AM	9/27/21 5:00 PM		Wenxuan Yan;Haowei Li;Ru...
2	✓	Project topic discussion	5 days	9/28/21 8:00 AM	10/4/21 5:00 PM	1	Wenxuan Yan;Haowei Li;Ru...
3	✓	Project initiation	5 days	10/5/21 8:00 AM	10/11/21 5:00 PM	2	Wenxuan Yan;Haowei Li;Ru...
4	✓	Project Proposal	13 days	10/12/21 8:00 AM	10/28/21 5:00 PM	3	Wenxuan Yan;Haowei Li;Ru...
5	✓	Problem Definition	13 days	10/12/21 8:00 AM	10/28/21 5:00 PM	3	Wenxuan Yan;Haowei Li;Ru...
6	✓	Project Objective	13 days	10/12/21 8:00 AM	10/28/21 5:00 PM	3	Wenxuan Yan;Haowei Li;Ru...
7	✓	Stakeholders List	13 days	10/12/21 8:00 AM	10/28/21 5:00 PM	3	Wenxuan Yan;Haowei Li;Ru...
8	✓	Success/Acceptance Criteria	13 days	10/12/21 8:00 AM	10/28/21 5:00 PM	3	Wenxuan Yan;Haowei Li;Ru...
9	✓	Use case diagram	3 days	11/1/21 8:00 AM	11/3/21 5:00 PM	4	Runze Zhang
10	✓	Selected Use case Descriptio	2 days	11/4/21 8:00 AM	11/5/21 5:00 PM	4	Runze Zhang
11	✓	Sequence diagram	2 days	11/4/21 8:00 AM	11/5/21 5:00 PM	9	Yichen Wang
12	✓	Class diagram	3 days	11/8/21 9:00 AM	11/11/21 9:00 AM	9	Simo Wu
13	✓	State-machine diagram	3 days	11/8/21 9:00 AM	11/11/21 9:00 AM	9	Simo Wu
14	✓	Subsystem Decomposition	3 days	11/8/21 9:00 AM	11/11/21 9:00 AM	9	Haowei Li
15	✓	ER – Diagram	2 days	11/12/21 9:00 AM	11/16/21 9:00 AM	14	Haowei Li
16	✓	GitHub link to your project sc	10 days	11/17/21 9:00 AM	12/1/21 9:00 AM	12	Wenxuan Yan;Haowei Li
17	✓	Project management	60 days	9/21/21 8:00 AM	12/13/21 5:00 PM		Wenxuan Yan;Haowei Li;Ru...
18	✓	WBS	1 day	12/9/21 9:00 AM	12/10/21 9:00 AM	16	Wenxuan Yan
19	✓	Task Assignment Matrix	1 day	12/9/21 9:00 AM	12/10/21 9:00 AM	16	Wenxuan Yan
20	✓	Gant chart of your project	1 day	12/9/21 9:00 AM	12/10/21 9:00 AM	16	Wenxuan Yan
21	✓	Task tracker	1 day	12/9/21 9:00 AM	12/10/21 9:00 AM	16	Wenxuan Yan
22	✓	Conclusion	1 day	12/10/21 9:00 AM	12/13/21 9:00 AM	18	Runze Zhang
23	✓	Reference	1 day	12/10/21 9:00 AM	12/13/21 9:00 AM	18	Runze Zhang
24	✓	powerpoint	0.9 days	12/2/21 9:00 AM	12/3/21 8:12 AM	16	Wenxuan Yan;Haowei Li;Ru...
25	✓	youtube	1 day	12/3/21 8:12 AM	12/6/21 8:12 AM	24	Wenxuan Yan;Haowei Li;Ru...

Figure 15.1 Task Assignment Matrix

Gantt Chart:

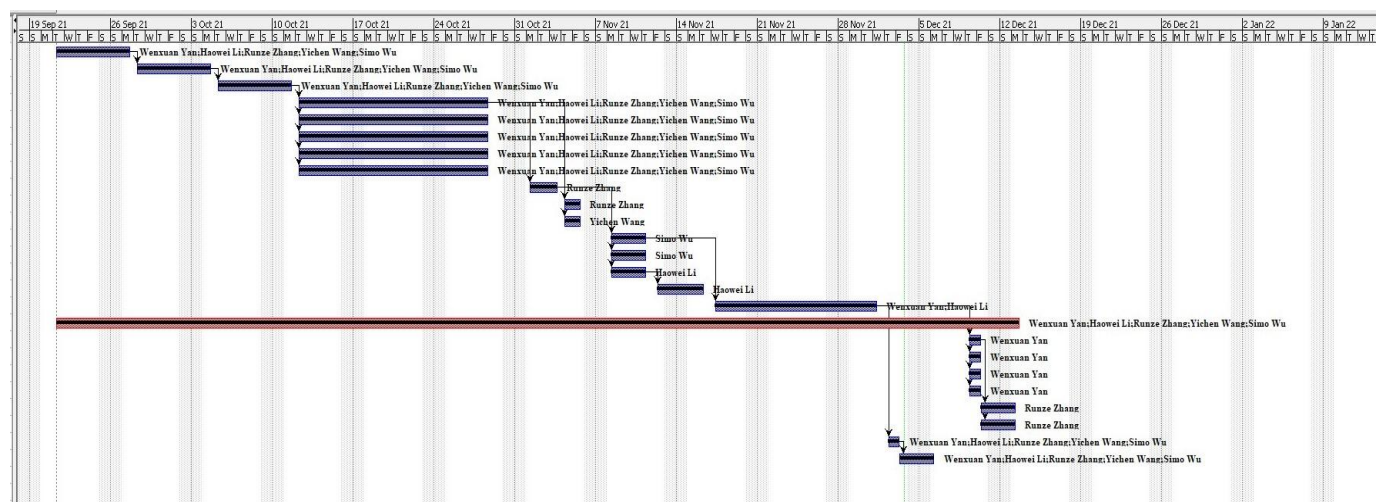


Figure 15.2 Gantt Chart

Task Tracker:

1	TASK LIST					
2	My TASKS	START DATE	DUE DATE	% COMPLETE	Member	
3	Project team formation	2021/9/21	2021/9/27	100%	5	
4	Project topic discussion	2021/9/28	2021/10/4	100%	5	
5	Project initiation	2021/10/5	2021/10/11	100%	5	
6	Project Proposal	2021/10/12	2021/10/28	100%	5	
7	Stakeholders List	2021/10/12	2021/10/28	100%	5	
8	Problem Definition	2021/10/12	2021/10/28	100%	5	
9	Project Objective	2021/10/12	2021/10/28	100%	5	
10	Success/Acceptance Criteria	2021/10/12	2021/10/28	100%	5	
11	Use case diagram	2021/11/1	2021/11/3	100%	1	
12	Selected Use case diagram	2021/11/4	2021/11/5	100%	1	
13	Sequence diagram	2021/11/4	2021/11/5	100%	1	
14	Class diagram	2021/11/8	2021/11/11	100%	1	
15	State-Machine diagram	2021/11/8	2021/11/11	100%	1	
16	Subsystem Decomposition	2021/11/8	2021/11/11	100%	1	
17	ER-Diagram	2021/11/12	2021/11/16	100%	1	
18	GitHub link	2021/11/17	2021/12/1	100%	2	
19	Project Management	2021/9/21	2021/12/6	100%	5	
20	WBS	2021/12/2	2021/12/3	100%	1	
21	Gantt chart	2021/12/2	2021/12/3	100%	1	
22	Task Tracker	2021/12/2	2021/12/3	100%	1	
23	Task Assignment Matrix	2021/12/2	2021/12/3	100%	1	
24	Conclusion	2021/12/3	2021/12/4	100%	1	
25	Reference	2021/12/3	2021/12/4	100%	1	
26	Powerpoint	2021/12/4	2021/12/6	100%	5	
27	Youtube	2021/12/4	2021/12/6	100%	5	

Figure 16.1 Task Tracker Diagram

Test Cases

Test Case ID	Test Case Description	Test Steps	Test Data	Expect Result	Actual Result	Test Result
T01	Add a car to the car list	1. Run the program 2. Choose "1. Add a car " 3. Enter the attributes required	Carid = 001 Name = Benz Type = SUV Level = Luxury Color = Red Insurance = Yes Price = \$100	Benz is successfully added to the carlist	Benz is successfully added to the carlist	Pass
T02	Delete a car from the car list	1. Run the program 2. Choose "8. Delete a car" 3. Enter the attributes required	Carid = 001	Deleted! This is car list:	Deleted! This is car list:	Pass
T03	Customer register to the customer list	1. Run the program 2. Choose "2. Add a customer(Register)" 3. Enter the attributes required	Name = Tester Phone number = 6178888888 ID = 111 Bank name = boa Account number = 999 Account type = checking	(111 Tester) is successfully added to the customer list	(111 Tester) is successfully added to the customer list	Pass
T04	Delete a customer from the customer list	1. Run the program 2. Choose "9. Delete a customer" 3. Enter the attributes required	User ID = 111	Deleted! This is customer list:	Deleted! This is customer list:	Pass
T05	The customer rent a car through the system	1. Run the program 2. Choose "3. Rent a car" 3. Enter the attributes required	User ID = 111 Carname = Benz	Benz is successfully rented	Benz is successfully rented	Pass
T06	The customer	1. Run the program	User ID = 111	111	111	Pass

	returns a car through the system	2. Choose "4. Return a car" 3. Enter the attributes required	Carname = Benz Return date = 2021/12/07	successfully returned Benz 2021/12/07 Revenue is: \$100,	successfully returned Benz 2021/12/07 Revenue is: \$100,	
T07	Show rental book	1. Run the program 2. Choose "5. Check rentalbook"	None	Show rental book	Show rental book	Pass
T08	Show car list	1. Run the program 2. Choose "6. Check carlist"	None	Show car list	Show car list	Pass
T09	Show customer list	1. Run the program 2. Choose "7. Check customer list"	None	Show customer list	Show customer list	Pass
T10	Exit the system	1. Run the program 2. Choose "10. To end the service"	None	Now the service will shut off System stopped.	Now the service will shut off System stopped.	Pass

Work Partition

- Problem Definition - Yichen Wang, Simo Wu
- Project Objective - Runze Zhang
- Stakeholders List - Runze Zhang
- Use case diagram(s) - Runze Zhang
- Selected Use case Descriptions - Yichen Wang, Runze Zhang, Haowei Li
- Sequence diagram(s) - Yichen Wang
- Class diagram(s) - Simo Wu
- State-machine diagram - Simo Wu
- Subsystem Decomposition - Haowei Li
- ER – Diagram - Haowei Li
- Code - Haowei Li, Wenxuan Yan, Runze Zhang, Yichen Wang, Simo Wu
- Test Cases - Runze Zhang
- Demo - Yichen Wang
- Project management - Haowei Li, Wenxuan Yan, Runze Zhang, Yichen Wang, Simo Wu
- WBS - Wenxuan Yan
- Task Assignment Matrix - Wenxuan Yan
- Gantt chart of your project - Wenxuan Yan
- Task tracker - Wenxuan Yan
- Conclusion - Simo Wu

Conclusion

The car rental system will make the rental process easier and more straightforward for college students with lower requirements of car rental users. It will reduce customer service costs by reducing labor costs and providing cheaper cars for the customers who do not have enough budget. The car rental system contains a customer system, car system, order system, and revenue system, which will perfectly work together and simplify the rental process to the largest extent.

The system could be used long-term since we could also add more features like VR video car viewing and customer experience evaluation. It is a system that can be continuously improved to have unlimited development possibilities to become more in line with users' needs and easier to use.

During the whole project, team members kept cooperating with each other and solved many difficult problems. For example, we first need to have a macro concept of the entire project, think about potential stakeholders, and realize the rental process shown in the diagram with codes. We divided the works to make it more efficient and followed up in real-time to ensure we were staying on track.

This project is of great help to the study of Software Engineering. From the conception of the project to the diagrams and then to code, we sort out what we learned in the whole semester and become more logical in the process of completing the project step by step.

Reference

Professor Ibrahim's lecture slides