**Deliverable 02-Analysis**

**Loza Tadesse, Amir Kemal, Nahome Hailemichael, Yanet Woldemichael**

Contents

[**Executive Summary** 3](#_Toc84875810)

[Section 1: Strategies and Techniques for Gathering Requirements 3](#_Toc84875811)

[Section 2: Determine Requirements 3](#_Toc84875812)

[Section 3: Use Cases 4](#_Toc84875813)

[Section 4: Activity Diagram 4](#_Toc84875814)

[Section 5: Class Diagram 4](#_Toc84875815)

[Section 6: Sequence Diagram 4](#_Toc84875816)

[**Strategies and Techniques for Gathering Requirements** 5](#_Toc84875817)

[**Determine Requirements** 8](#_Toc84875818)

[**Use Cases** 8](#_Toc84875819)

[**Activity Diagram** 13](#_Toc84875820)

[**Class Diagram** 14](#_Toc84875821)

[**Sequence Diagram** 15](#_Toc84875822)

# **Executive Summary**

According to the U.S. Census Bureau, a rural area is any area that is neither an urbanized area (50,000 or more people) nor an urban cluster (between 2,500 to 50,000 people). Currently, over 60 million (1 in 5) Americans live in such areas. A small town with a college campus could likely support a small bikeshare system or other forms of shared mobility, whereas a small town within a rural setting may require more creative strategies given the lower population density. For this system, we will have the context to be in Madison, SD. This system will combat the transportation issues for the town, provide adequate access of mobility, affordable transportation options, and create an income for the university. For the long run, we believe that the system will attract more students who would think the lack of transportation is an issue.

## Section 1: Strategies and Techniques for Gathering Requirements

This first section of the report explains the requirements-gathering techniques to collect information; requirements analysis strategies drive the kind of information that is gathered and how it is ultimately analyzed. The requirements analysis strategies and requirements gathering happen concurrently and are complementary activities.

## Section 2: Determine Requirements

The second section is to determine the purpose of requirements. It turns the very high-level explanation of the business requirements stated in the system request into a more precise list of requirements that can be used as inputs to the rest of analysis (creating functional, structural, and behavioral models). This expansion of the requirements ultimately leads to the design of the system.

## Section 3: Use Cases

The third section is the use cases which provide a bird’s-eye view of the business processes contained in the evolving system. The use-case diagram depicts the communication path between the actors and the system. Use cases and their use-case description documentation tend to be used to model both the contexts of the system and the detailed requirements for the system.

## Section 4: Activity Diagram

The fourth section is the activity diagrams, which are used to model the behavior in a business process independent of objects.

## Section 5: Class Diagram

The fifth section is the class diagram is a static model that shows the classes and the relationships among classes that remain constant in the system over time. The class diagram depicts classes, which include both behaviors and states, with the relationships between the classes.

## Section 6: Sequence Diagram

The sixth section is the sequence diagram which is a dynamic model that shows the explicit sequence of messages that are passed between objects in a defined interaction. Because sequence diagrams emphasize the time-based ordering of the activity that takes place among a set of objects, they are very helpful for understanding real-time specifications and complex use cases.

# **Strategies and Techniques for Gathering Requirements**

For the requirement gathering step, the basic process of analysis is divided into three steps: understanding the as-is system, identifying improvements, and developing requirements for the to-be system. For this project, we will be skipping the first step because there is no current system that exists on which we are building on. The second step would be identifying improvements for the system. For this system, one of the strategies used to gather requirements is the root cause analysis. In this analysis, we focus on problems, not solution. If any possible, root causes are identified for several problems; those should be investigated first because there is a good chance that they are the real root causes influencing the symptom problems. For our system, some possible root causes are:

* + Scanners on the bus don’t work.
  + The system does not display the most accurate and current bus schedules and paths.
  + Problem with the data management and cause misunderstanding between user and system.
  + Have inaccurate route schedule caused by lack of efficient algorithm from the information received by uses.

The other strategy is information benchmarking. We looked at Uber and Lyft which are both huge transportation systems that use GPS to track the location of their vehicles and see how far away the vehicle is from the customers. We plan to implement this feature to track the route and movements of the buses so that users can have a more transparent view of the system. Other features we used was that when users request a driver, the car which is most close to the user would be alerted for the users request. This feature can be used for our system to direct our customers to the closes stop from their current location by calculating the distance from their current location from the other bus stops and displaying the shorted distance possible.

The Techniques we used to gather the information for our system was the interview method and the JAD method. For the interview, we first created and arranged question that were closed and open-ended question. After that, we used the top to bottom approach because we wanted to interview at random starting with broad question to act as ice breakers then eventually working towards more specific ones. Finally, we decided to do a random interview around campus to get the most accurate, random, and authentic perspective students and faculty had on the idea of our project. Most students responded well on the idea and gave us some ideas we needed to add to include to our system. One of the new ideas we got was to include debit card payment for people who don’t have an ID card and are not resident of Madison like parents or friends who came to visit but need access to the minibus transportation system so included a debit card onetime payment option in our system.

The second technique we used that seemed fit for our system was the JAD (joint application system). The main reason we used this technique was because all team members are international students so brainstorming for easy and effective solution to find a transportation system in the school seemed convenient. The system we are building will try to solve issues most international students face. First, all team members discussed on features we want to system to have and listed the requirements that were reasonable and built on that to better the system we were building.

# **Determine Requirements**

The purpose of requirements determination is to turn the very high-level explanation of the business requirements stated in the system request into a more precise list of requirements that can be used as inputs to the rest of analysis (creating functional, structural, and behavioral models). This expansion of the requirements ultimately leads to the design of the system. A functional requirement relates directly to a process a system must perform or information it needs to contain. Non-functional requirements refer to behavioral properties that the system must have, such as performance and usability.

**Functional requirements**

1. **View schedule**

* Users see the traveling schedule

1. **Update schedule**

* Drivers update schedule

1. **Reserve Van**

* Users track where van is located

1. **Manage Reservation**

* Users can make reservation for specific occasions

1. **Make payment**

* Non- subscribed users must make payment to access the buses

1. **Fill out information**

* Non- subscribed users must fill out information to make payment

1. **Authenticate Schedule**

* Admin must authenticate every schedule agreement between drivers and users

**Nonfunctional requirements**

1. **Operational requirements**

* The system will operate in IOS, Android, and as a webapp.
* The system should be able to connect to the Internet.
* The system should be able to access DSU’s cloud server.
* The system will develop a route using AI by gathering the most wanted routes.

1. **Performance requirements**

* The system will be able to update the van’s location every 3 second.
* The system will recalculate the estimated arriving time at a location every 5 second.

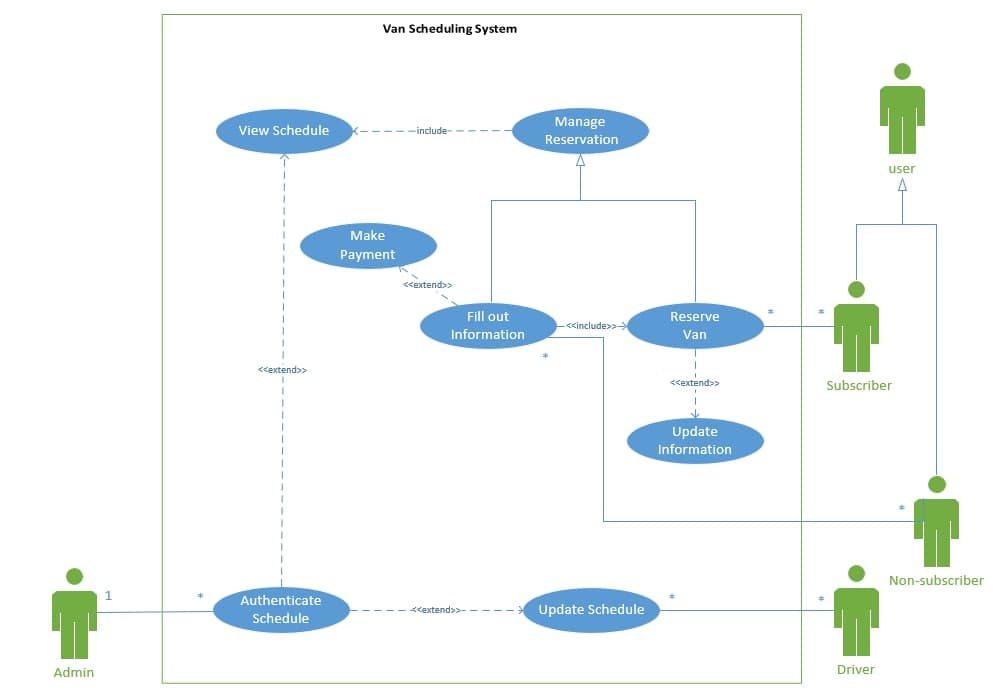
1. **Security requirements**

* Only authenticated users can use the system.

1. **Culture and political requirements**

* No special cultural and political requirements are anticipated.

# **Use Cases**

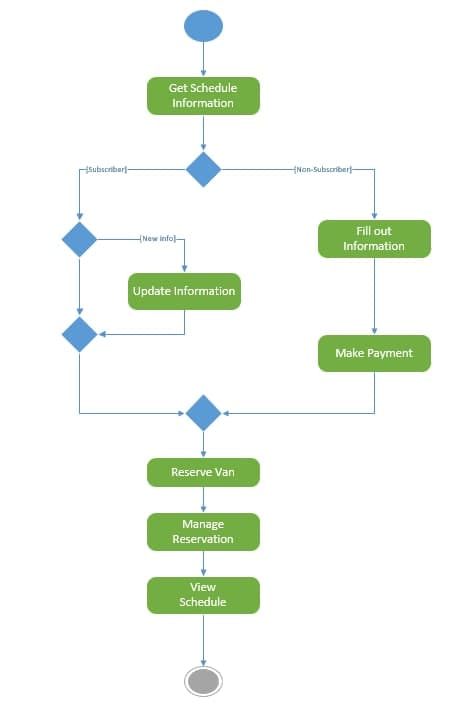


**Use Cases Description**

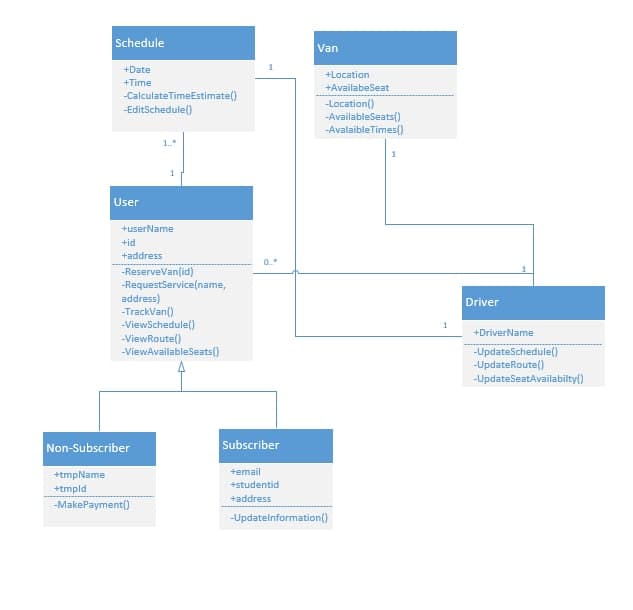
|  |  |  |  |
| --- | --- | --- | --- |
| Use Case Name: Reserve Van | | ID:1 | Importance Level: medium |
| Primary Actor: User | Use Case Type: Essential | | |
| Stakeholders and Interests:  User: Wants to manage reservation and schedule a van for transport. | | | |
| Brief Description: This use case shows describes a user can schedule a van transport service as well as updating. | | | |
| Trigger: User calls and requests a service from the driver then asks to choose a from available vans.  Type: External | | | |
| Relationships:  Association: Subscribed User, Unsubscribed User  Include: Fill out Information  Extend: Update Information  Generalization: Manage Reservation | | | |
| Normal Flow of Events:   1. User contacts and requests the driver using the system to schedule a transport 2. If the User information has changed   Execute the Update Information use case   1. If the User is unsubscribed   Execute the Fill out Information use case  Execute the Make Payment use case   1. The user requests to see schedule information 2. The system provides schedule availability 3. The user reserves an available van 4. The system provides schedule confirmation | | | |
| SubFlows:   1. Modify Van reservation    1. The driver asks the user for new schedule times    2. The system stores updated time and van schedule 2. Cancel Van reservation    1. The driver asks the user for confirmation number    2. The system removes the schedule and marks that van as available | | | |
| Alternate/Exceptional Flows:  S-1,2a 1: The system could not confirm schedule  S-1,2a 2:The driver is unavailable | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case Name: Fill Out Information | | ID:1 | Importance Level: medium |
| Primary Actor: User | Use Case Type: Essential | | |
| Stakeholders and Interests:  User: Wants to view the schedule of the van. | | | |
| Brief Description: This use case provides a form that a user can fill in with their information | | | |
| Trigger: User needs reserve a van without subscription  Type: External | | | |
| Relationships:  Association: Non-subscriber  Include: Reserve Van  Extend: Make Payment  Generalization: Manage Reservation | | | |
| Normal Flow of Events:   1. Non-subscriber contacts and requests the driver using the system to schedule a transport 2. The system requests the non-subscriber to fill out necessary information 3. Non-subscriber fills out information 4. The system asks the non-subscriber to make payment 5. If the non-subscriber has completed payment   Execute the Reserve Van use case   1. The system stores schedule information and sends confirmation | | | |
| SubFlows:  None. | | | |
| Alternate/Exceptional Flows:  3a. The non-subscriber wants to become a subscriber  4a. The system can’t process the payment | | | |

# **Activity Diagram**



# **Class Diagram**



# **Sequence Diagram**

