

VIETNAM NATIONAL UNIVERSITY,
HCMC UNIVERSITY OF TECHNOLOGY
FACULTY OF COMPUTER SCIENCE AND ENGINEERING



SOFTWARE ENGINEERING (CO3001)

URBAN WASTE COLLECTION AID

UWC 2.0

Task 1: Requirement elicitation

Professors

Bùi Hoài Thắng & Nguyễn Đức Anh

Members

Trần Phạm Minh Đăng	2052070
Ngô Trương Trọng Nghĩa	2053264
Đinh Xuân Quang	2053359
Nguyễn Hoàng Thuận	2052729
Nguyễn Trần Quốc	2053384

Ho Chi Minh City, February 2023

Contents

1.1 Describe the domain context of Urban waste management in Vietnam. Who are relevant stakeholders? What are their current needs? In your opinion, what benefits UWC 2.0 will be for each stakeholder?	3
1.1.1 Domain context in Vietnam.....	3
1.1.2 Relevant stakeholders and their current needs	4
1.1.3 Benefit.....	4
1.2 Describe all functional and non-functional requirements that can be inferred from the project description. Draw a general use-case diagram for the whole system.....	5
1.2.1 Functional requirements:	5
1.2.2 Non-functional requirements:	6
1.2.3 System use-case diagram:.....	7
1.3 For the Task assignment module, draw its use-case diagram and describe the use-case using a table format.	8
1.3.1 Task assignment use-case diagram:.....	8
1.3.2 Use-case scenario.....	9

1.1 Describe the domain context of Urban waste management in Vietnam. Who are relevant stakeholders? What are their current needs? In your opinion, what benefits UWC 2.0 will be for each stakeholder?

1.1.1 Domain context in Vietnam

The volume of home garbage in metropolitan regions likewise tends to expand in an upward direction, based on the economic development and particularly the current population of the entire world, including Vietnam. Statistics show that Vietnam produces more than 27.8 million tons of waste annually from a variety of sources, with more than 46% of this waste coming from urban sources such homes, restaurants, marketplaces, and enterprises. The main cities in Vietnam are Hanoi, Ho Chi Minh City (HCMC), Hai Phong, Da Nang, and Can Tho, which account for 70% of all trash produced in the nation. With a population of 12 million, Ho Chi Minh City is the biggest urban area in Vietnam. Ho Chi Minh City, a megacity and the nation's economic hub, is reportedly faced with an increasing volume of rubbish each year—an estimated 7200–7800 tons every day.

Despite the enormous amount of waste, Ho Chi Minh City's waste management system is still restricted to manual local collection models and being processed in external landfills, like other underdeveloped nations. In addition, low wealth levels in developing nations impede technological advancement, leading to subpar garbage collection services. The lack of a support application to optimize routes and garbage collection schedules is most notable, since it results in an uneven collection schedule that overloads waste processing facilities and drives up transportation expenses.

The goal of this project is to develop a management system for professional waste management services in order to increase efficiency and optimize resources in urban garbage collection. Meanwhile, there is a system called Urban Waste Collection Aid 1.0 (UWC 1.0) that has limited functionality and poor performance, thus an enhanced version of this system is needed to manage the waste collection problem faster and better. As a result, organization X is contracted to develop an information management system called UWC 2.0 in order to improve efficiency of garbage collection of Service provider Y. .

1.1.2 Relevant stakeholders and their current needs

A stakeholder is defined as a person or group of persons who might impact or are affected by a certain project; the relevant stakeholders in this project include:

- **Back officers:** Who operate UWC 2.0 to assign tasks, create calendar, coordinate front collectors and janitors and determine the routes and vehicles to be used. **Their needs** are a good system to manage schedules and send messages to collectors and janitors to divide their work calendar, and vehicles. The system also allows to assign vehicles, tasks, and created optimized routes for collectors. Moreover, it can keep and retrieve information on all MCPs and their capacity.
- **Janitors:** Who gather rubbish in their allocated areas and transport it to the Major Collecting Points (MCPs) using trollers. **Their current needs** are a system that tells them how many MCPs they need to travel to gather rubbish and notifies them when the MCPs are completely loaded. Moreover, Janitors can then divide themselves into small groups to gather waste more rapidly. Furthermore, they may determine how many trollers they will need to bring with them to collect.
- **Collectors:** Who drive different types of vehicles will pick up garbage from all janitors at an MCPs. During his working shift, one collector drives only one vehicle. **Their current needs** are have an overview of their work calendar and detail of their task on a daily and weekly basis. The system can notified about the MCPs if they are fully loaded, check in/out their task everyday, communicate with other janitors, collectors and back officers and also declare the whole distance they must go each day in order to predict how much gasoline will be spent and pay for it in order to inform managers afterwards.

1.1.3 Benefit

- Back officers will have more tasks to complete, but they will save time and money while increasing work performance since they do it more logically.
- Collectors and janitors having the correct spot to do their job faster and directly effect on the amount of trash they gather
- Because of the efficiency of UWC 2.0, the amount of trash will be 70% less than before and the air will be fresher and street will be cleaner after all

1.2 Describe all functional and non-functional requirements that can be inferred from the project description. Draw a general use-case diagram for the whole system.

1.2.1 Functional requirements:

- General functional requirements:
 - + The system should allow communication between back officers, collectors, janitors and customers
 - + The system should be usable from a mobile device, a tablet device or a normal computer/ laptop.
 - + Users can message other users using user ID and the system should notify the user whenever they receive a message.
 - + In the future, the system should be able to be used in a variety of Republic Services.
- Back officer:
 - + Have an overview of tasks coordinate front janitors and collectors (dashboard)
 - + View workers's information (including their work calendar).
 - + Have an overview of vehicles and their technical details (weight, capacity, fuel consumptions, etc).
 - + Communicating with janitors and collectors.
 - + Have an overview of all MCPs's information (location, capacity).
 - + Assign vehicles to janitors and collectors.
 - + Assign janitors and collectors to MCPs (task) .
 - + Create a route for each collector. Assigned route is optimized in terms of fuel consumption and travel distance.
- Collector and janitor:
 - + Have an overview of their work calendar.
 - + Have a detailed view of their task on a daily and weekly basis.
 - + Communicating with other janitors , collectors and back officers.
 - + Be notified about the MCPs if they are fully loaded.
 - + Can view the map and the route for each collector.
 - + Check in / check out tasks every day.

1.2.2 Non-functional requirements:

- Security:
 - + Accounts should have passwords longer than 12 characters
 - + Users can only try logging 3 times, after 3 fails, the account is locked.
 - + At any given time, a user will be limited to logging onto only one device.
- Performance:
 - + The system should be able to handle real-time data from at least 1000 MCPs at the moment and 10.000 MCPs in five years
 - + The messages should be communicated in a real-time manner with delay less than 2 second.
 - + Have a detail view of their task on a daily and weekly basic. All important information should be displayed in one view (without scrolling down).
 - + UWC 2.0 system interfaces should be in Vietnamese, with an opportunity to switch to English in the future.
- Availability:
 - + Information should be updated from MCPs every 15 minutes with the availability of at least 95% of their operating time.
 - + The system must works 24/7.
- Reliability:
 - + System failure rate (crashing) must not exceed 0.05%.
 - + UWC 2.0 is expected to import and to use the existing data from UWC 1.0. It is expected that the Task Management to be inter-operable with the UWC 1.0 as much as possible.
 - + Automatic data backup with a frequency of 1 time / 2 week in case of problems and emergencies.
 - + The app should be updated on a regular basis.

1.2.3 System use-case diagram:

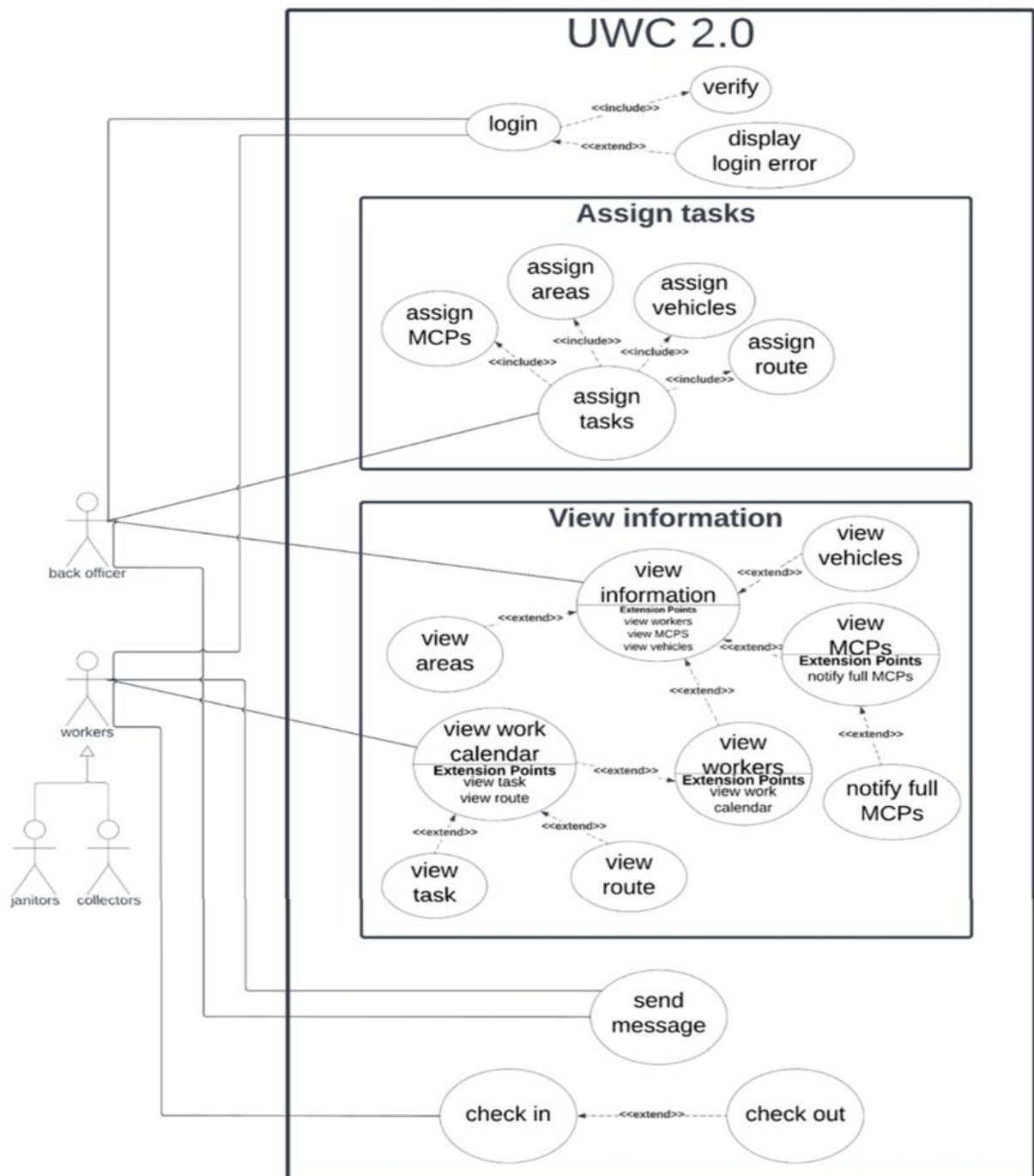
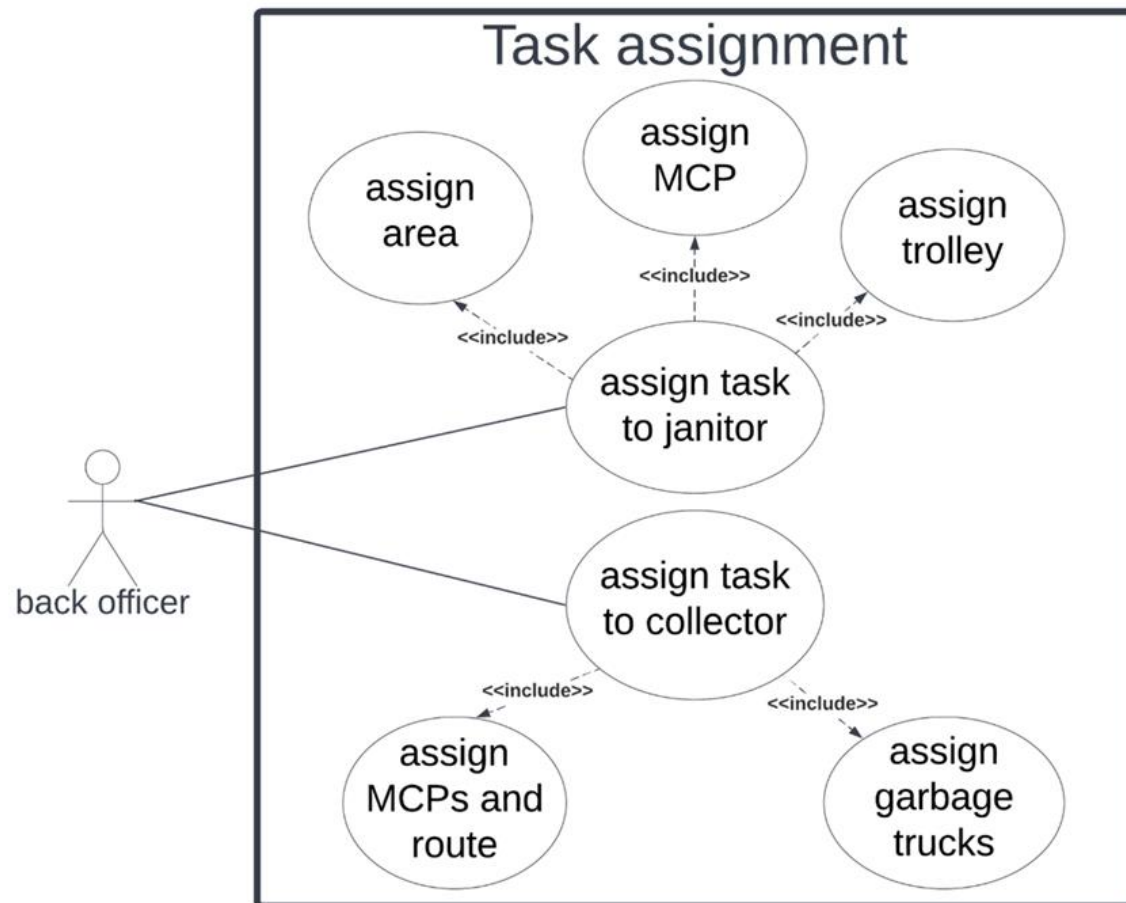


Figure 1: Use case diagram for UWC 2.0 system

Based on functional requirements, the UWC 2.0 system has 4 actors (back officers, janitors, collectors, workers - generalized from collector and janitor) and 2 primary function (task assignment, information management)

1.3 For the Task assignment module, draw its use-case diagram and describe the use-case using a table format.

1.3.1 Task assignment use-case diagram:



1.3.2 Use-case scenario

Assign task to janitor

Use - case name:	Assign task to janitor
Use - case overview:	Back officer assigns task to janitor
Actors:	Back officer
Pre conditions:	<ul style="list-style-type: none">- Devices must connect to the Internet.- User must be a back officer
Trigger:	Click on the “View” button on a Janitor from the list of Janitors.
Steps:	<ul style="list-style-type: none">- System shows information of that janitor, including ID, name and their tasks info.- Back officer clicks on the Calendar icon button.- System shows the calendar.- Back officer chooses a day from the calendar and click “Set date”.- The system shows the Tasks for that day.- Back officer starts assigning for the janitor.- System saves the record.
Post condition:	- Back officer successfully assigns area for janitor.
Exception flow:	None

Assign task to collector

Use - case name:	Assign task to collector
Use - case overview:	Back officer assigns task to collector
Actors:	Back officer
Pre conditions:	<ul style="list-style-type: none">- Devices must connect to the Internet.- User must be a back officer.
Trigger:	Click on the “View” button on a Collector from the list of Collectors.
Steps:	<ul style="list-style-type: none">- System shows information of that collector, including ID, name and their tasks info.- Back officer clicks on the Calendar icon button.- System shows the calendar.- Back officer chooses a day from the calendar and click “Set date”- The system shows the Tasks for that day.- Back officer starts assigning for the janitor- System saves the record.
Post condition:	Back officer successfully assigns area for janitor.
Exception flow:	None

Assign area

Use - case name:	Assign area
Use - case overview:	Back officer assigns area to janitor
Actors:	Back officer
Pre conditions:	<ul style="list-style-type: none">- Devices must connect to the Internet.- User must be a back officer.
Trigger:	Click on “Update” button on Area row on janitor task
Steps:	<ul style="list-style-type: none">- System shows a list of available areas.- Back officer chooses an area.- System shows that the area has been selected.- Back officer click “Add”- System adds that MCP to the task, saves the record, creates the optimal route between the new area and MCP, and shows the result to the Back officer
Post condition:	Back officer successfully assigns area for janitor.
Exception flow:	None

Assign MCP

Use - case name:	Assign MCP
Use - case overview:	Back officer assigns MCP to janitor
Actors:	Back officer
Pre conditions:	<ul style="list-style-type: none">- Devices must connect to the Internet.- User must be a back officer.
Trigger:	Click on “Update” button on MCP row on janitor task
Steps:	<ul style="list-style-type: none">- System shows a list of available MCPs.- Back officer chooses an MCP.- System shows that the MCP has been selected.- Back officer click “Add”- System adds that MCP to the task, saves the record, creates the optimal route between the MCP and area, and shows the result to the Back officer.
Post condition:	Back officer successfully assigns MCP for janitor.
Exception flow:	None

Assign trolley

Use - case name:	Assign trolley
Use - case overview:	Back officer assigns trolley to janitor
Actors:	Back officer
Pre conditions:	<ul style="list-style-type: none">- Devices must connect to the Internet.- User must be a back officer.
Trigger:	Click on “Update” button on Trolley row on janitor task
Steps:	<ul style="list-style-type: none">- System shows a list of available trolleys.- Back officer chooses a trolley.- System shows that the trolley has been selected.- Back officer click “Add”- System adds that trolley to the task, saves the record, and shows the result to the Back officer.
Post condition:	Back officer successfully assigns trolley for janitor.
Exception flow:	None

Assign MCPs and route

Use - case name:	Assign MCPs
Use - case overview:	Back officer assigns MCPs to collector
Actors:	Back officer
Pre - condition:	<ul style="list-style-type: none">- Devices must connect to the Internet.- User must be a back officer.
Trigger:	Click on “Update” button on “Chosen MCPs” row on janitor task
Steps:	<ul style="list-style-type: none">- System shows a list of available MCPs.- Back officer chooses an MCP..- System shows that the MCP has been selected.- Back officer click “Add”- System adds that MCP to the list of MCPs, saves the record, and creates the optimal route between MCPs in the order they were added to the list. and shows the result to the Back officer.
Post condition:	Back officer successfully assigns MCP for janitor.
Exception flow:	None

Assign garbage trucks

Use - case name:	Assign <i>vehicle</i>
Use - case overview:	Back officer assigns <i>vehicle</i> to collector
Actors:	Back officer
Pre - condition:	<ul style="list-style-type: none">- Devices must connect to the Internet.- User must be a back officer.
Trigger:	Click on “Update” button on <i>Vehicle</i> row on collector task
Steps:	<ul style="list-style-type: none">- System shows a list of available vehicles.- Back officer chooses a vehicle.- System shows that the vehicle has been selected.- Back officer click “Add”- System adds that vehicle to the task, saves the record, and shows the result to the Back officer.
Post condition:	Back officer successfully assigns vehicle for collector.
Exception flow:	None