

VIET NAM NATIONAL UNIVERSITY HO CHI MINH CITY
HO CHI MINH UNIVERSITY OF TECHNOLOGY



SOFTWARE ENGINEERING
ASSIGNMENT – TASK 1

Author:

Name	Student ID	Class
Nguyễn Nam Kha	2052515	CC03
Nguyễn Ngọc Hòa	2052485	CC03
Hồ Nguyễn Ngọc Bảo	2052036	CC03
Phan Mai Tấn Lợi	2052158	CC03
Đào Quang Vinh	2053586	CC03

Ho Chi Minh city, 09/2022

Contents

1. Introduction.....	1
2. Requirement elicitation	2
2.1. Stakeholders	2
2.2. User story	2
2.3. User requirements analysis	3
3. Functional and non-functional requirements	5
3.1. Functional requirements.....	5
3.2. Non-functional requirements	6
4. Use-case diagram of whole system.....	8
5. Task assignment	9
5.1. Use case diagram.....	9
5.2. Use case scenario	10
5.2.1. Assign task to janitors.....	10
5.2.2. Assign task to collectors	11
5.2.3. Modify tasks	12
5.2.4. Assign trolleys	13
5.2.5. Assign garbage trucks.....	14
5.2.6. Assign area and MCP	15
5.2.7. Assign MCPs and route.....	16

1. Introduction

Urban waste management is one of several significant problems faced by many countries in the world and thus considered one of the important points to be improved in Sustainable Development Goal (SDG) 11: sustainable cities and communities and SDG 6: clean water and sanitation. The continuous increase of population growth factor and accompanied by the consumptive lifestyle is one of the main reasons for the high volume of human's waste, especially in the urban areas of developing countries. The main problem in the management of urban waste occurs due to the imbalance between the production and the capability to manage it, particular attention is given to developing countries that continue to prioritize development and economic growth. At the moment, solid waste management in the urban area is proved to be costly as well as ineffective so both the government and organizations continuously emphasize the importance of having an effective waste management system in order to improve the life quality of citizens, societies and environments.

This project aims to provide a management system for the professional waste management services in order to improve their efficiency and optimize their resources in urban waste collection. In the meantime, there exists a system called Urban Waste Collection Aid 1.0 (UWC 1.0) with limited features and poor performance so there is a demand for the upgraded version of this system to handle the waste collecting problem faster and better. Therefore, a new system called UWC 2.0 will be developed to solve the remaining problem of UWC 1.0.

2. Requirement elicitation

2.1. Stakeholders

By definition, a stakeholder is a person or group of people who can affect or be affected by a given project, the relevant stakeholders in this project are:

- Back officer: Operating UWC 2.0 to assign tasks, create calendar and coordinate front collectors and janitors.
- Janitor: Manually collect garbage from Major Collecting Points (MCPs) based on the task assigned to them in UWC 2.0.
- Collector: Picking up garbage at MCPs based on the task assigned to them in UWC 2.0.

2.2. User story

- As a back officer, I want to have a list or dashboard of my collector and janitors so that I can control them more easily.
- As a back officer, I want to chat to my janitors and collectors when needed.
- As a back officer, I want to have more MCPs so that I can assign tasks more flexibly.
- As a back officer, I want to see whether a MCP is full or not
- As a back officer, I want to see if a collector or janitor is absent or not automatically.
- As a back officer, I want to know if a vehicle is suitable for my collector's task.
- As a back officer, I want to control all work calendars of collectors and janitors.
- As a back officer, I want to assign tasks to janitor and collector for a whole week.
- As a foreign back officer, I want the language to be English so that I can understand more clearly
- As a janitor, I want to see clearly important details of my task.
- As a janitor, I want to check my attendance easily.
- As a janitor, I want to know if my assigned MCP is full so I don't have to go there to dump my garbage.
- As a janitor, I want to chat with other janitors or back officers when needed.
- As a janitor, I want to see what I am going to do this week.
- As a foreign janitor, I want the language to be English so I can read the task.
- As a collector, I want to know what I am going to drive.
- As a collector, I want to know which MCP I have to pick up garbage.
- As a collector, I want to save fuel and my time.
- As a collector, I want to check my attendance easily.
- As a collector, I want to communicate with other collectors or janitors and back officers.
- As a collector, I want to see my task this week.
- As a foreign collector, I want the language to be English so I can read the task.

2.3. User requirements analysis

From all the user stories, we can understand more clearly about user requirements. Therefore, we shall address their problems and their needs correctly.

Problem:

- Don't know MCPs capacity
- Have small amount of MCPs
- Route assigned to collectors is not appropriate
- Back officer cannot view multiple collectors and janitors in one view.
- Have no information about technical details of vehicles so the back officer can assign an unsuitable vehicle for a collector
- Not able to send messages inside the application but have to use another message application.
- Important information about the work calendar may be displayed in separate places.
- Have to manually text the back officer to check in or check out and the back officer has to manually write down the working status.
- Janitors deliver garbage to a full capacity MCP so they have to find another MCP to dump garbage.
- Collectors and janitors can only view the task daily without knowing tomorrow's task.

Need:

- An in-app message service to send messages between users in real-time manner.
- Have information about MCPs capacity and are updated frequently.
- Back officer needs a dashboard or a list to view multiple janitors or collectors
- System can handle large amount of MCPs
- Predetermined route is optimized to save fuel and time.
- Show detailed information of vehicles.
- Have a built-in check in or check out function
- Display important information of work calendar in one view.
- Receive notification when assigned MCP is full.
- Collectors and janitors need a weekly view work calendar.

Benefit:

- Collectors can save time, resources and increase productivity since their route is optimal.
- Managing a large amount of MCPs means that they can collect garbage from more locations so it will improve the efficiency.
- Check in/Check out function will reduce the errors made by back officers in case they forget to write it down.
- Collectors and janitors can see the important details of their task in one view, this will create ease of use.
- Back officers can assign appropriate vehicles to collectors in terms of their route so they can travel more efficiently.
- More convenient to use in-app messages based on User ID.
- Back officers save time in finding collectors or janitors to assign work by using a dashboard.
- Improve efficiency by assigning collectors to appropriate MCPs based on its capacity.

3. Functional and non-functional requirements

3.1. Functional requirements

General functional requirements

- Each user must log in and be uniquely identified by user ID provided by the company.
- Users can message other users using user ID and the system should notify the user whenever they receive a message.
- The system will automatically send a message to all the workers assigned to a fully loaded MCPs.

Back - officers shall be able to:

- Assign collecting areas, MCPs and trolleys to the janitors.
- Assign MCPs, predetermined routes and garbage trucks to the collectors.
- Create the optimal route for collectors with the input is multiple MCPs.
- Modify tasks.
- View workers and can search or sort workers based on name, user ID, location of area or MCPs assigned to them or status check in/ check out.
- View a worker's work calendar and details about each working shift like assigned MCPs, vehicles, routes.
- View MCP's name, capacity, location and can search or sort MCPs based on the given attributes
- View vehicles and their technical details and can search based on their user ID assigned to.

Janitors shall be able to:

- View their work calendar and details about each working shift like assigned collecting areas, MCPs, trolleys.
- Check in/check out and get recorded the amount of working time by system.

Collectors shall be able to:

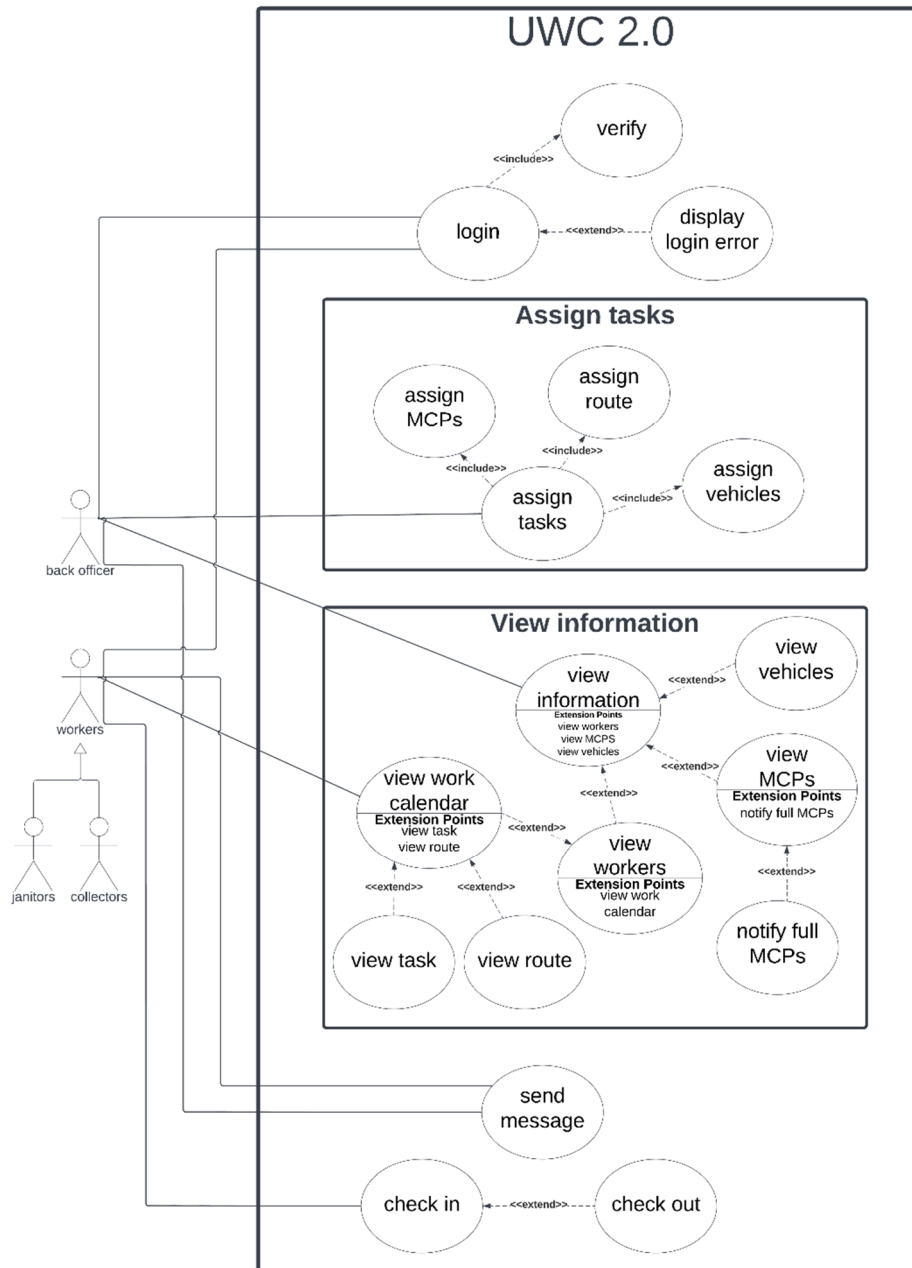
- View their work calendar and details about each working shift like assigned MCPs, garbage trucks, optimal routes.
- Check in/check out and get recorded the amount of working time by system.

3.2. Non-functional requirements

1. Security	<ul style="list-style-type: none">- Web page should have protection against DDoS attacks- Users can only try logging 3 times, after 3 fails, the account is locked.- Accounts should have passwords longer than 12 characters.
2. Performance	<ul style="list-style-type: none">- System should take less than 5 seconds to load.- System response should be less than 1 second for each request.- Data transmission between users should take less than 3 seconds.- The system can handle real-time data from at least 1000 MCPs now.- Server can handle at least 1000 simultaneous connections.
3. Usability	<ul style="list-style-type: none">- Collectors/janitors can use the software proficiently after 10 minutes of tutorials.- Back officers can use the software proficiently after 15 minutes of tutorials.- All major functions must be accessible in up to 3 screen clicks/taps.
4. Integrity	<ul style="list-style-type: none">- Tasks assigned to workers and information delivered to back officers must be 100% accurate.
5. Availability	<ul style="list-style-type: none">- System should be available 24/7, downtime shall not exceed 10 seconds in any day.- Information should be updated from MCPs every 15 minutes with the availability of at least 95% of their operating time.

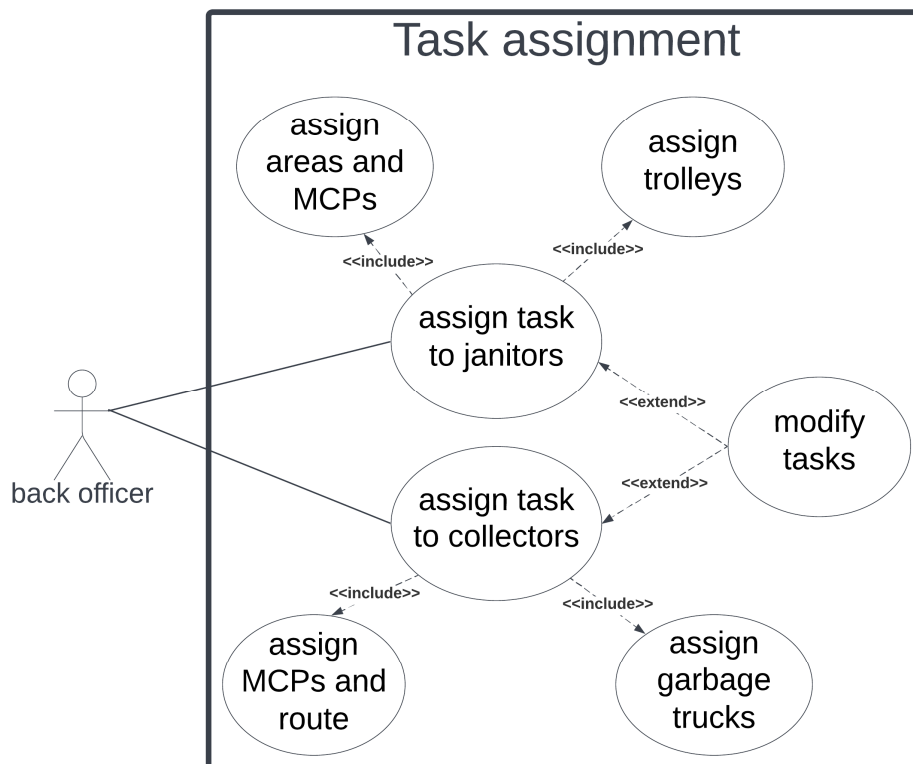
6. Reliability	<ul style="list-style-type: none">- System failure rate (crashing) must not exceed 0.05%.- Failed request rate must not exceed 0.03%.- Probability of data corruption must not exceed 1%.- The amount of time between two critical failures is more than a month.
7. User interface	<ul style="list-style-type: none">- All important information should be displayed in one view without scrolling down- Minimize horizontal scrolling.- Lists must be displayed in alphabetical order by default.
8. Supportability	<ul style="list-style-type: none">- The system should be compatible with mobile devices running iOS or Android and various web browsing applications such as Chrome, Firefox, Safari for laptops and desktops.- System interface should be responsive to many screen sizes such as mobile phones, tablets, laptops,...
9. Scalability	<ul style="list-style-type: none">- The system can be extended to handle real-time data from 10.000 MCPs in five years.- System can be updated to newer versions in the future with the capability to be interoperable as well as using the existing data from older versions.
10. Localization	<ul style="list-style-type: none">- System interface supports Vietnamese, with the development to switch to English in the future.- Date must be written in format DD/MM/YYYY.
11. Maintainability	<ul style="list-style-type: none">- Routine maintenance for the system should take less than 30 minutes.

4. Use-case diagram of whole system



5. Task assignment

5.1. Use case diagram



5.2. Use case scenario

5.2.1. Assign task to janitors

Use - case name:	Assign task to janitors
Description:	Back officer assigns task to janitors
Actor:	Back officers
Trigger:	Click on “Assign work”
Pre - condition:	<ul style="list-style-type: none">- Device must connect to the Internet- User must be a back officer.
Post - condition:	Back officer successfully assigns work to janitors.
Normal flow:	<ol style="list-style-type: none">1. System shows a list of janitors.2. Back officer chooses a janitor to assign work.3. System shows a table of 7 days.4. Back officer chooses one day to assign work for that day.5. System shows 3 options: Assign area, Assign MCP, Assign trolley.6. Back officer assigns these 3 options to the janitor in turn.7. Back officer clicks the “Save” button.8. System records the assignment.
Alternative flow:	<ol style="list-style-type: none">2.a. Back officer searches for the desired janitor.<ol style="list-style-type: none">2.a.1. System shows the list of janitors after searching.2.a.2. Back officer chooses the desired janitor to assign work. <p>Use case continues from step 3.</p>
Exceptions:	<ol style="list-style-type: none">2.b. System cannot find the desired janitor. System shows a blank list.

5.2.2. Assign task to collectors

Use - case name:	Assign task to collectors
Description:	Back officer assigns task to collectors
Actor:	Back officers
Trigger:	Click on “Assign work”
Pre - condition:	<ul style="list-style-type: none">- Device must connect to the Internet- User must be a back officer.
Post - condition:	Back officer successfully assigns work to collectors.
Normal flow:	<ol style="list-style-type: none">1. System shows a list of collectors.2. Back officer chooses a collector to assign work.3. System shows a table of 7 days.4. Back officer chooses one day to assign work for that day.5. System shows 3 options: Assign MCPs, Assign garbage truck, Assign route.6. Back officer assigns these 3 options to the collector in turn.7. Back officer clicks the “Save” button.8. System records the assignment.
Alternative flow:	<ol style="list-style-type: none">2.a. Back officer searches for the desired collector.<ol style="list-style-type: none">2.a.1. System shows the list of collectors after searching.2.a.2. Back officer chooses the desired collector to assign work. Use case continues from step 3.
Exceptions:	<ol style="list-style-type: none">2.b. System cannot find the desired collector . System shows a blank list.

5.2.3. Modify tasks

Use - case name:	Modify tasks
Description:	Back officer modifies tasks for workers.
Actor:	Back officers
Trigger:	Click on a day on the worker's work calendar.
Pre - condition:	<ul style="list-style-type: none">- Devices must connect to the Internet.- User must be a back officer.- Chosen day has already been assigned tasks.
Post - condition:	Back officer successfully modifies tasks for workers.
Normal flow:	<ol style="list-style-type: none">1. System pops up an alert saying "Tasks have already been assigned for this day. Do you wish to modify tasks?" with two options Yes and No.2. Back officer chooses Yes.3. System shows the options to modify: areas, MCPs, vehicles, routes.4. Back officer modifies the desired options.5. Back officer clicks the "Save" button.6. System records the change.
Alternative flow:	
Exceptions:	<ol style="list-style-type: none">2.a. Back officer chooses No.<ol style="list-style-type: none">2.a.1. Use case stops.

5.2.4. Assign trolleys

Use - case name:	Assign trolleys
Description:	Back officer assigns trolleys to janitors
Actor:	Back officers
Trigger:	Click on “Assign trolleys”
Pre - condition:	<ul style="list-style-type: none">- Device must connect to the Internet- User must be a back officer.
Post - condition:	Back officer successfully assigns trolleys for janitors.
Normal flow:	<ol style="list-style-type: none">1. System shows a list of trolleys.2. Back officers choose the desired trolley.3. System records the assignment.
Alternative flow:	<ol style="list-style-type: none">2.a. Back officer searches for the desired trolley.<ol style="list-style-type: none">2.a.1. System shows the trolley after searching.2.a.2 Back officer chooses the desired trolley. Use case continues from step 3.
Exceptions:	<ol style="list-style-type: none">2.c. System cannot find the desired trolley. System shows a blank list.2.d. Back officer chooses the trolley that has already been assigned to another janitor.<ol style="list-style-type: none">2.d.1. System pops up an alert saying “This trolley has already been assigned to another janitor”.

5.2.5. Assign garbage trucks

Use - case name:	Assign garbage trucks
Description:	Back officer assigns garbage trucks to collectors
Actor:	Back officers
Trigger:	Click on “Assign garbage trucks”
Pre - condition:	<ul style="list-style-type: none">- Device must connect to the Internet- User must be a back officer.
Post - condition:	Back officer successfully assigns garbage trucks for collectors.
Normal flow:	<ol style="list-style-type: none">1. System shows a list of garbage trucks.2. Back officers choose the desired garbage trucks.3. System records the assignment.
Alternative flow:	<ol style="list-style-type: none">2.a. Back officer searches for the desired garbage trucks.<ol style="list-style-type: none">2.a.1. System shows the garbage truck after searching.2.a.2. Back officer chooses the desired garbage truck. Use case continues from step 3.
Exceptions:	<ol style="list-style-type: none">2.c. System cannot find the desired garbage truck. System shows a blank list.2.d. Back officer chooses the garbage truck that has already been assigned to another collector.<ol style="list-style-type: none">2.d.1. System pops up an alert saying “This garbage truck has already been assigned to another collector”.

5.2.6. Assign area and MCP

Use - case name:	Assign area and MCP
Description:	Back officer assigns area and MCP to janitor
Actor:	Back officers
Trigger:	Click on “Assign MCPs and route”
Pre - condition:	<ul style="list-style-type: none">- Device must connect to the Internet- User must be a back officer.
Post - condition:	Back officer successfully assigns area and MCP for collectors.
Normal flow:	<ol style="list-style-type: none">1. System shows a list of areas.2. Back officer clicks on an area.3. System shows information about that area (name, location, MCP) with options “Add to task” or “Close”4. Back officer clicks “Add to task”.5. System adds that area to the task.6. Back officer clicks “Save”.7. System records the assignment.
Alternative flow:	<p>2.a. Back officer searches for the desired areas. 2.a.1. System returns the list of areas after searching. 2.a.2. Back officer chooses the desired area. Use case continues from step 3.</p> <p>4.a. Back officer clicks “Close”. Use case continues from step 1.</p>
Exceptions:	2.c. System cannot find the desired area. System shows a blank list.

5.2.7. Assign MCPs and route

Use - case name:	Assign MCPs and route
Description:	Back officer assigns MCPs and route to collectors
Actor:	Back officers
Trigger:	Click on “Assign MCPs and route”
Pre - condition:	<ul style="list-style-type: none">- Devices must connect to the Internet.- User must be a back officer.
Post - condition:	Back officer successfully assigns routes for collectors.
Normal flow:	<ol style="list-style-type: none">1. System shows a list of MCPs.2. Back officer clicks on a MCP.3. System shows information about that MCP (name, location, capacity) with options “Add to task” or “Close”4. Back officer clicks “Add to task”.5. System adds that MCP to the task.6. Back officer clicks “Create route”.7. System creates the optimal route between MCPs in the order they were added to the route.8. Back officer clicks “Save”.9. System records the assignment.
Alternative flow:	<p>2.a. Back officer searches for the desired MCP.</p> <p>2.a.1. System returns the list of MCPs after searching.</p> <p>2.a.2. Back officer chooses the desired MCP.</p> <p>Use case continues from step 3.</p> <p>4.a. Back officer clicks “Close”.</p> <p>Use case continues from step 1.</p> <p>6.a. Back officer wants to add more MCPs to the task.</p> <p>Use case continues from step 1.</p>
Exceptions:	2.c. System cannot find the desired MCP. System shows a blank list.