VIETNAM NATIONAL UNIVERSITY,

HO CHI MINH CITY UNIVERSITY OF TECHNOLOGY

FACULTY OF COMPUTER SCIENCE AND ENGINEERING

SOFTWARE ENGINEERING

Assignment

“URBAN WASTE COLLECTION – UWC 2.0”

Group: 4

MEMBERS: Phạm Hoàng Minh - 2152771

Tăng Tuấn Đạt – 2152512

Tạ Gia Khang – 2152652

Nguyễn Chánh Tín - 2153043

Table of Contents:

Chapter 1: **Requirement elicitation**

**1.1: Relevant stakeholders and the benefits of UWC 2.0**

**1.2: Functional and non-functional requirements:**

**1.3 Task assignment use-case diagram**

**Chapter 1: Requirement elicitation**

**1.1: Relevant stakeholders and the benefits of UWC 2.0**

As a developing country, urban waste management is one of Vietnam's biggest challenges, especially in Ho Chi Minh city, where industrialization has led to a massive population density of 4375(1) people in a km2. In addition, Ho Chi Minh city is riddled with small streets, underdeveloped roads, which means the waste disposal process is different from the first world countries. Big dumpster trucks cannot fit such roads, so the city has to resort to janitors with trollers to collect the garbage, then they will transport it to a Major Collecting Point (MCP), where collectors can collect using heavy vehicles and move them to a facility in the outskirts area. Attempts have been made to optimize such a system, noticeably the Urban waste collection aid or UWC 1.0, which helps to organize and centralize the procedure. However, as the amount of waste continue to grow, a remodeled version of UWC is more necessary, a new project called UWC 2.0.

The stakeholders of the UWC 2.0 project are back officers, collectors and janitors with the addition of a service provider (may from an internal or external source). All of the stakeholders are in need of a centralized system to help with communication, viewing the work calendar and checking-in at the start and the end of each shift/day. There are also special needs for each type of position. A back officer needs a way to create and manage the work calendar, the workflow of janitors and collectors. Janitors and collectors, on the other hand, need a way to view the said work calendar and the work assigned. This system needs to be fast and effective with a shallow learning curve to allow everyone to migrate from the previous system. In addition, a service provider or administrator needs an ability to easily transfer any data from the previous system and easily help any of the other employees in case technical problems arise.

UWC 2.0 will benefit all stakeholders by improving the efficiency and effectiveness of the waste collection process and providing real-time information and communication among stakeholders. It will also satisfy all of the needs of every stakeholder with an easy to follow with an easy-to-follow guide and user-friendly interface to help all workers to use this website in less than an hour. They can also save time by using the map to find the nearest non-full MCP for janitors or help collectors make informed decisions about their route by purposefully skipping MCPs that are not full. All of this can be viewed and monitored by a back officer and in case of emergency, they can contact other workers.

**1.2: Functional and non-functional requirements:**

1.2.1: Functional requirements:

Back Officers:

- As a back officer, I want to have an overview of work calendar of janitors and collectors

- As a back officer, I want to have an overview of MCPs and capacity

- As a back officer, I want the MCPs' availability to be updated every 15 minutes

- As a back officer, I want to assign tasks weekly, vehicles and routes monthly

- As a back officer, I want to send messages/notices to individual or in general

-As a back officer, I want to send an announcement to a group of people

- As a back officer, I want to create route for collectors

- As a back officer, I want to assign area to janitors

Janitors:

- As a janitor, I want to have an overview of work calendar daily or weekly in one view

- As a janitor, I want to check out/mark my progress

- As a janitor, I want waste site to be near each other

- As a janitor, I want to check-in and check-out at the start and end of every shift

- As a janitor, I want to send messages/notices to others

Collectors:

- As a collector, I want the optimal route among MCPs

- As a collector, I want to have an overview of the routes.

- As a collector, I want to check-in and check-out at the start and end of every shift

- As a collector, I want to view information about my assigned vehicle

- As a collector, I want to view the status of MCPs in my route

- The system allows employees to send messages to others

1.2.2: Non-functional requirements

- The system shall support seamless import and use of data from UWC 1.0.

- The system shall support inter-operability with UWC 1.0.

- The system shall ensure navigation with delays of no more than 1.5 seconds.

- The system shall ensure message delays of no more than 1 second.

- The system shall handle real-time data from at least 1000 MCPs at the moment and 10000 MCPs in five years

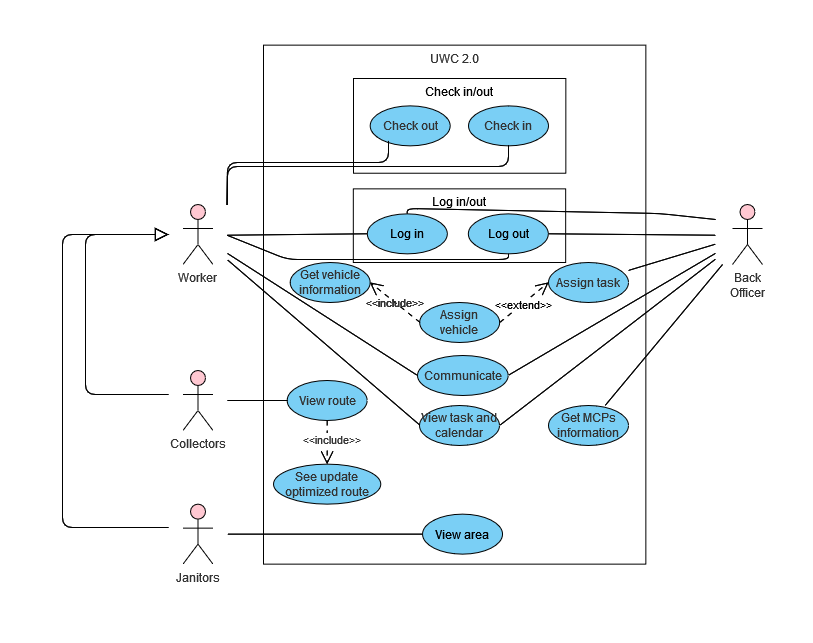
- The system shall support Vietnamese with the possibility of English in the future.

- The system shall require a maximum of two hours of training for users to operate.

- The system shall guarantee 99% up-time and limit downtime to less than 5 seconds per occurrence.

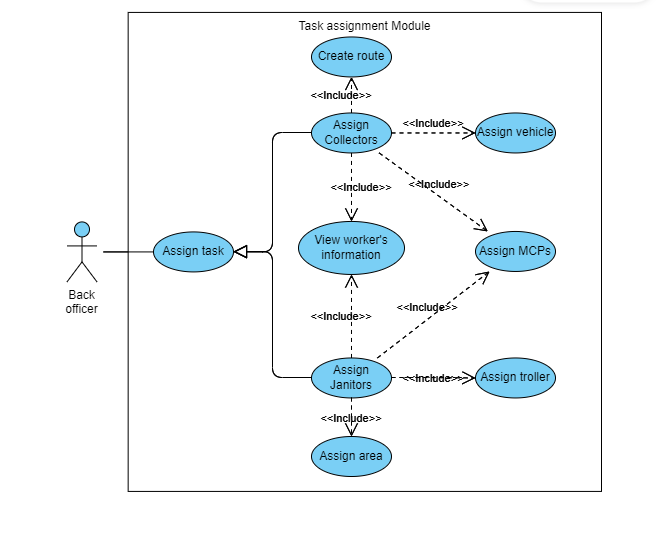
- The system shall ensure sanitization of all inputs.

1.2.3: General use-case diagram:



**1.3 Task assignment use-case diagram and table descriptions**:

1.3.1: Use case diagram:



1.3.2: Table descriptions:

Collector

|  |  |
| --- | --- |
| Use-case name | Assign MCPs. |
| Use-case overview | To check and assign suitable MCPs to workers. |
| Actors | Back officers. |
| Preconditions | 1. The system is running.  2. The database is connected to MCPs.  3. Internet connection is available. |
| Trigger | Users click the “Assign MCPs” button. |
| Steps | 1. Retrieve all MCP’s information from the database.  2. Display a list containing suitable MCPs which have analyzed vacancy after previous assignment.  3. Overview the MCPs which the users click and ask if they are chosen.  4. Update the chosen ones to the worker’s schedule weekly. |
| Post conditions | Update the database and display the worker’s schedule on the screen. |
| Exception flow | If there are no available MCPs, return to the homepage and send a notification to users. |

|  |  |
| --- | --- |
| Use-case name | Assign vehicles. |
| Use-case overview | To check and assign suitable vehicles to collectors. |
| Actors | Back officers. |
| Preconditions | 1. The system is running.  2. The database is connected to vehicle storage.  3. Internet connection is available. |
| Trigger | Users click the “Assign vehicles” button. |
| Steps | 1. Retrieve all vehicles' information from the database.  2. Display a list containing suitable vehicles which have good status and are near to the working area of the collector.  3. Overview the vehicles which the users click and ask if they are chosen.  4. Update the chosen ones to the worker’s monthly schedule. |
| Post conditions | Update the database and display the worker’s schedule on the screen. |
| Exception flow | If there are no available vehicles, return to the homepage and send a notification to users. |

|  |  |
| --- | --- |
| Use-case name | Create a route. |
| Use-case overview | Create a suitable working route for the collectors. |
| Actors | Back officers. |
| Preconditions | 1. The system is running.  2. The database is connected to map and MCPs.  3. Internet connection is available.  4. Should have at least one MCP in the schedule of the collector. |
| Trigger | Users click the “Create a route” button. |
| Steps | 1. Retrieve all MCP’s information from the database.  2. Compute and find routes for the collector which has no conflict with the others and optimized in terms of fuel consumption and travel distance.  3. Display the routes on the screen, suggest the best one.  4. Overview the routes which the users click and ask if they are chosen.  5. Update the chosen ones to the worker’s weekly schedule. |
| Post conditions | Update the database and display the worker’s schedule on the screen. |
| Exception flow | If there are no available routes, return to the homepage and send a notification to users. |

Janitor

|  |  |
| --- | --- |
| Use-case name | Assign task |
| Use-case overview | To assign tasks to either janitors or collectors |
| Actors | Back officers |
| Preconditions | 1. The system is running  2. Internet connection is available  3. The back officer has login |
| Trigger | Back officer click the “Assign task” button |
| Steps | 1. Give option to display list of either janitors or collectors  2. Retrieve data on the database  3. Display the list of choice on the screen  4. Back officer goes through steps:  “view worker’s information”, “assign area”, “assign troller” and “assign MCPs”  5. A confirmation message pops up on screen  6. The back officer presses to confirm |
| Post conditions | The task is assigned to the corresponding individual and other information is updated to the database. A message is sent to notice the chosen individual |
| Exception flow | None |

|  |  |
| --- | --- |
| Use-case name | View worker’s information |
| Use-case overview | To view worker’s general information before assigning task to them |
| Actors | Back officers |
| Preconditions | 1. Back officer has chosen either assign task to janitor or collector  2. The list of employees has shown on the screen  3. Back officer has determined the individual to receive task |
| Trigger | Back officer click on the individual’s name on the list |
| Steps | 1. Retrieve their information from the database  2. Display the chosen individual’s availability and their work calendar  3. Update the information on screen every 15 minutes in case other back officers has just assigned task to the janitor |
| Post conditions | Required information is displayed on the screen of users’ devices and is easy to read |
| Exception flow | 1. The worker is unavailable for the day (with permission)  2. Display the unavailable status on screen  3. Hide option to assign task  4. The back officer tap the left arrow on top left to go back to previous list |

|  |  |
| --- | --- |
| Use-case name | Assign area |
| Use-case overview | To assign area for janitor to do their work |
| Actors | Back officers |
| Preconditions | 1. The janitor is available for the day in “view worker’s information”  2. The back officer has chosen the janitor to receive task |
| Trigger | Users click the “Assign area” after viewing the janitor’s availability |
| Steps | 1. Retrieve the map with all the areas need cleaning from the database  2. Display the information on the user screen  3. Update the area in colors that show status: has none janitors, has more than 2 janitors, has more than 4 janitors  4. Back officer choose the suitable area and assign task |
| Post conditions | The task is selected for the janitor |
| Exception flow | None |

|  |  |
| --- | --- |
| Use-case name | Assign troller |
| Use-case overview | To assign troller to janitor for their work |
| Actors | Back officers |
| Preconditions | 1. Back officer has chosen a janitor to receive task  2. Back officer has assigned the working area to the janitor |
| Trigger | Users has chosen the working area for the janitor |
| Steps | 1. Retrieve all the trollers’ information  2. Pop up a window for the back officer to select troller from  3. The list is sorted according to the nearest distance to the chosen working area  4. The back officer choose a troller for the janitor |
| Post conditions | A troller is selected for the janitor |
| Exception flow | 1. Out of available trollers  2. Abort the assign task, area of the janitor  3. Return back to the main screen |