Urban waste collection aid - UWC 2.0

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. Particular attention is given to developing countries that continue to prioritize development and economic growth. In urban context, solid waste management is costly and ineffective. Improvement of waste collection and management is emphasized by governments and organizations for positive impacts on cities, societies and environments.

Waste collection is often designated to an organization that provides professional waste management services. A typical waste collection process involves (1) back officers, who operate a central system to create calendar, coordinate front collectors and janitors ,(2) collectors, who drive different types of vehicles and (3) janitors who manually collect garbage from Major Collecting Points (MCPs). Calendar and tasks were assigned among teams of janitors and coordinated by back officers. These assignments are often arranged in a weekly basic. Back officers also plan which vehicles to use and their routes. This planning activity happens every month. Everyday, the back officers sent messages with information about collecting route and time to collectors and janitors. Janitors use trollers (see Figure 1b) to collect garbage in their assigned areas and deliver to the MCPs. Collectors will pick up garbage from all janitors at an MCP. One collector drives only one vehicle during his working shift. The collector will drive through several MCPs with a predetermined route by back officers.

Figure 1a. Collecting vehicles

Figure 1b. Trollers

 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. The solution will include a Task Management module that allows:

Back officers to:

1. Have an overview of janitors and collectors, their work calendar
2. Have an overview of vehicles and their technical details (weight, capacity, fuel consumptions, etc)
3. Have an overview of all MCPs and information about their capacity. Information should be updated from MCPs every 15 minutes with the availability of at least 95% of their operating time.
4. Assign vehicles to janitors and collectors
5. Assign janitors and collectors to MCPs (task)
6. Create a route for each collector. Assigned route is optimized in term of fuel consumption and travel distance.

7. Be able to send message to collectors and janitors

Collectors and janitors to:

1. Have an overview of their work calendar
2. 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).
3. Be able to communicate with collectors, other janitors and back officers. The messages should be communicated in a real-time manner with delay less than 1 second.
4. Check in / check out task every day
5. Be notified about the MCPs if they are fully loaded

There are some constraints to the development of UWC 2.0. There exists a current system UWC 1.0 with a database. UWC 2.0 is expected to import and to use the existing data from UWC1.0. It is expected that the Task Management to be inter-operable with the UWC 1.0 as much as possible. 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. UWC 2.0 system interfaces should be in Vietnamese, with an opportunity to switch to English in the future.

---------Task ----------

Task 1: Requirement elicitation	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? Guide: At least three paragraphs have to be written (1) a paragraph about domain context, (2) a paragraph about stakeholders and their needs, (3) a paragraph about the benefits of UWC2.0 for each stakeholder. The section has to be *understandable*, refer to *reliable* sources and information has to be *justified*.
	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 Guide: At least 05 functional requirements for each stakeholders. Requirements to be written as single sentences. The requirements must be *complete, unambiguous, consistent and correct*. The use case diagram has to be *complete and correct syntactically*.
	1.3	For the Task assignment module, draw its use-case diagram and describe the use-case using a table format Guide: one use case diagram shall be drawn. The use case diagram has to be *complete and correct syntactically*. One use-case description table for each use case using the example from the lecture. The number of steps in an use-case should be more than 3. At least one use case should have an Exception flow. Example:

		Use-case ID	U4.
		Use-case name	Overview MCP.
		Use-case overview	To provide information about Major Collecting Points (MCPs) and their current capacity.
		Actors	Back officers.
		Preconditions	1. The system is running. 2. Database is connected to MCPs. 3. Internet connection is available.
		Trigger	Users click the "MCPs' overview" button.
		Steps	1. Retrieve all MCPs' information and capacity. 2. Display all information on the screen of users' devices. 3. Update MCPs' capacity every 15 minutes, then retrieve new capacities from the database and overwrite the old capacities with the new ones.
		Post conditions	Required information are displayed on the screen of users' devices and are updated every 15 minutes.
		Exception flow	None

Task 2: System modelling	2.1	Draw an activity diagram to capture the business process between systems and the stakeholders in Task Assignment module Guide: draw an activity diagram for each use case with a swimlane (https://circle.visual-paradigm.com/activity-diagram-example-swimlane/) for different stakeholders. The number of diagrams should be maximum 5. Try to use as many notations of the diagrams as possible (https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-activity-diagram/). Write 01 paragraph for each diagram to describe it. The diagrams should be *correct and complete*.
	2.2	Think about a possible way for a back officer to assign vehicles to janitors and collectors. Draw a sequence diagram to visualize this process Guide: draw 01 sequence diagram for each use case with a swimlane for different stakeholders. The number of diagrams should be maximum 5. Try to use as many notations of the diagram as possible (https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-sequence-diagram/) Write 01 paragraph for each diagram to describe it. The diagrams should be *correct and complete*.
	2.3	Draw a class diagram of Task Assignment module as comprehensive as possible Guide: draw 01 activity diagram for the whole module. Try to use as many notations of the diagram as possible (https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-class-diagram/). The diagram should be *correct and complete*.
	2.4	Develop MVP 1 as user interfaces of either a Desktop-view central dashboard for Task Management for back-officers OR a Mobile-view Task assignment for Janitors and Collectors. Decide yourself what to include in the view. Use a wireframe tool like Figma or Adobe XD, or Illustrator Guide: The wireframe shall be made from scratch. Language is Vietnamese. Any photos or materials taken from Internet has to give reference/ links. No implementation (backend or frontend) is needed at this stage. The user interface shall have *at least 05 screens*. There should be *clickable links* to navigate among the screen. The user interface should be as *close to a final software's screenshot as possible* (high fidelity wireframe).
Task 3: Architecture design	3.1	Use a layered architecture to design the UWC 2.0 system. Describe how will you present your User Interface. Describe how will you store your data. Describe how you will access to external services/ APIs.

		Guide: draw 01 architectural diagram for the overall design of UWC 2.0 system. Write 01 paragraph for your Presentation strategy, 01 paragraph for Data storage approach and 01 paragraph for API management. The architectural decisions in these paragraphs should be *justified* and *associated with external links* for detail approaches.
	3.2	Draw a component diagram for the Task Assignment module Guide: draw 01 component diagram for the whole module. Try to use as many notations of the diagram as possible (https://online.visual-paradigm.com/diagrams/tutorials/component-diagram-tutorial/). See the notation description from the lecture. Write 01 paragraph for each diagram to describe it. The diagram should be *correct and complete*.
Task 4: Implementation – Sprint 1	4.1	Setting up an online repository (github, bitbucket, etc) for version control. Guide: setup Github (https://docs.github.com/en/get-started/quickstart/hello-world) and a hello world example. Every team shall *have an account with public/ shareable links* to the repository.
	4.2	Adding documents, materials and folders for Requirement, System modelling and Architectural design. Use the selected version control system to report the changes to these files. Guide: the repository shall have a *readme file*. The repository shall be frequently maintained (*basing on last updated time).*
	4.3	Conducted a usability test with the user interface you developed in MVP1. Guide: follow the guideline https://www.nngroup.com/articles/usability-testing-101/ Key steps include: 1. Recruit participants/ testers. 2. Define tasks. 3. Define test strategy (qualitative vs. quantitative, remote vs. in-person. 4. Conduct the test. 5. Document the feedback from testers. All key steps shall be documented in 01 test report. The test report should be complete, understandable, having illustrative photos. The test report should be less than 10 A4 pages.
Task 5: Implementation – Sprint 2	5.1	Develop MVP2 with input from Task 2.4 and Task 4.3. You are free to choose the programming language (HTML, Javascript, Python, C#, etc). It is not required to implement a database in the backend. Data can be hard coded in code files. Guide: the MVP should be demonstrable in either a desktop view or a mobile view. The MVP should capture *most important value* for all stakeholders from the project description. The MVP should have *at least 05 different screens/ views*. There should be *hyperlinks to navigate through views*.
	5.2	Demonstrate the whole project from Task 1 to Task 5 Guide: a presentation slide shall be prepared. Each team should practice the demonstrations many times in advance. The presentation should be *straight to the point, contains lessons learned for the team*. The demonstration should be *prepared, correct, brief* and has *good quality*.