Web-based ERP System with GIS for Facilities Maintenance in Maritime Cargo Terminals

Moertini, V. S.,[1*] Mukhti, J. A.[2] and Setiawan, C.[2]
[1]Informatics Department, Parahyangan Catholic University, Indonesia, E-mail: moertini@unpar.ac.id
[2]Dinamaritama Konsultan Rekayasa, Bandung, Indonesia
**Corresponding Author*

Abstract

The civil facilities in maritime cargo terminals must be maintained to prevent uninterrupted services in handling cargo as well as to defer the stoppage of the whole port supply chain. These facilities include wharves, yards, warehouse and piping, and all the components attached to these which are needed to support the port operations. Given the vast area of the maritime cargo terminals (a terminal may consist of several berths of which the area is 10-100 ha each), identifying the exact location of each facility with its current condition information is important for the decision makers to design proper repair actions for the damaged parts of the facilities. Hence, GIS is needed. ERP frameworks offer advantages that can be adopted in facilities maintenance system. However, GIS is not yet supported by the frameworks. This paper presents the design of the proposed ERP system with GIS module and its implementation. To facilitate interactive maps and to display high quality of maps on web browsers, Scalable Vector Graphics (SVG) format is adopted. The system has passed the user acceptance test and is used by the largest maritime cargo terminals in Indonesia.

1. Introduction

Maritime port terminals or simply called as ports generally occupy a vast area. Their infrastructure consists of wharves, quays and jetties, breakwaters, roads and parks, buildings (including yards and warehouses) and dredging (INA, 1998). Some ports may also have pipings. Wharves, yards, and warehouses have many facility components to support their functions (for instance, the components of a wharf are fenders, bollards, and so on). By design, each of the components is "attached" to a certain location on a particular facility. Wharves, yards, warehouses, and piping are usually "divided" into smaller unit of areas (such as blocks with identification numbers). The structure of wharves, yards, warehouses, and piping as well as their facility components must be maintained regularly to ensure uninterrupted cargo services that can cause economic loss and to prevent major damages that also lead to loss (Zhang et al., 2017).

Given their large sizes, the structure of facilities may be damaged just on few separated "tiny spots". The decision makers, who usually do not visit the vast terminal areas, need to easily and quickly identify the exact location of these spots as well as the complete information of the damages (pictures and the description of the damages) in order to plan the repair actions. Likewise, the exact locations of broken/failed facility components (such as fenders, bollards and lights) also need to be identified prior to the replacement. For these reasons, GIS for facilities management is needed. Despite of this fact, any research result discussing this specific domain of GIS has not been found (most of information systems for marine terminals relate to supply-chain management systems, such as (Kia et al., 2000)). One work found, which is close to this domain, is WebGIS for maintaining coastal structures, particularly breakwaters (Maia et al., 2017).

It is known that the Enterprise Resource Planning (ERP) system that comes with a lot of useful modules for organizations (such as Accounting, CRM, HR, Marketing, MRP, Sale and many kinds of reporting) are needed by organizations (APTEAN, 2017), especially large corporations that manage marine terminals. With ERP, data organization can be captured in the real time, thus many kinds of reports can be generated based on the current data and can be displayed as texts as well as graphs (dash-boards). Web-based ERP frameworks offering modules that can be customized quickly (by developers) to meet organizations' need are available. However, to the best of our knowledge, there are currently no ERP framework that offers GIS modules which can be customized. This paper intends to contribute in designing a specific web-based ERP and GIS for maintaining civil facilities in large marine cargo terminals that can capture and process non-spatial as well as spatial data in the real time to produce text, graphs, as well as spatial reports. The case study is an organization handling many vast marine terminal cargo ports in Indonesia. As ERP design is quite common, in this paper the focus is on discussing how to enhance it with GIS features.

2. Related Literature Review

2.1 Maritime Port Terminals and Facilities Maintenance

Maritime port terminals are the most vital aspect of the national transportation infrastructure as they are used as trade facilitation. Given their multiple roles (such as exports and/or imports, mercantile trading center and distribution center), cargo ports should also be considered as one of the most important aspects of maritime transport (Alderton, 2008). Maritime cargo port terminals refer to a complete port facility used to accommodate ships loading and discharging and cargo stacking and handling onshore. The port accommodation consists of berthing accommodation and storage facilities. Berthing accommodation includes general cargo berths (wharves, quays, piers and docks), terminals, bulk cargo facilities, container and roll-on/roll-off terminals, liquefied gas terminals, and so on. Storage facilities include transit sheds (along the wharves or docks), back-up storage away from the dock, warehouses, stockyards and stacking areas for containers, stockpiles for bulk cargo, and so on. The cargo can be classified into (Petering, 2011): Dry bulk (such as salt, grain, minerals and coal byproducts), liquid bulk (such as crude oil, gasoline and liquefied natural gas), break bulk (such as steel, lumber and heavy machinery), automobile and containerized (such as finished consumer goods).

The total terminal area is covering (a) the apron or the area behind the berth front, (b) the primary yard area or container storage area, (c) the secondary yard area, which includes the entrance facility, parking, office buildings, customs facilities, container freight station, empty container storage, and container maintenance area. The apron, yard, and each of the facility require a wide area, which makes a marine terminal as a vast area of land and shore in result. The area required for a multi-purpose terminal will vary between about 5-15 ha/berth, and for a container terminal about 10-100 ha/berth, depending of the generation of the container ship. Few major facilities (Figure 1) that need to be maintained are described as follows:

(a) Wharf or quay or berth (in general): A structure attached to land to which a vessel is moored. It is either the area of water between or next to a human-made structure or group of structures involved in the handling of boats or ships, usually on or close to a shore, or the structures themselves. The structure can be constructed from timber, masonry, cement, or other material with sufficient depth of water to accommodate vessels and receive and discharge cargo or passengers. To support the operations of loading/discharge of ships, wharfs are equipped with fenders, bollards, berth meter, drainages, water pipe for vessels, electricity grid and so on (Petering, 2011)

(b) Yard: The storage area of primary and secondary yard can be constructed using concrete or paving blocks. The yards are usually equipped with crane rail, lightings, drainage, marks, signs and so on.

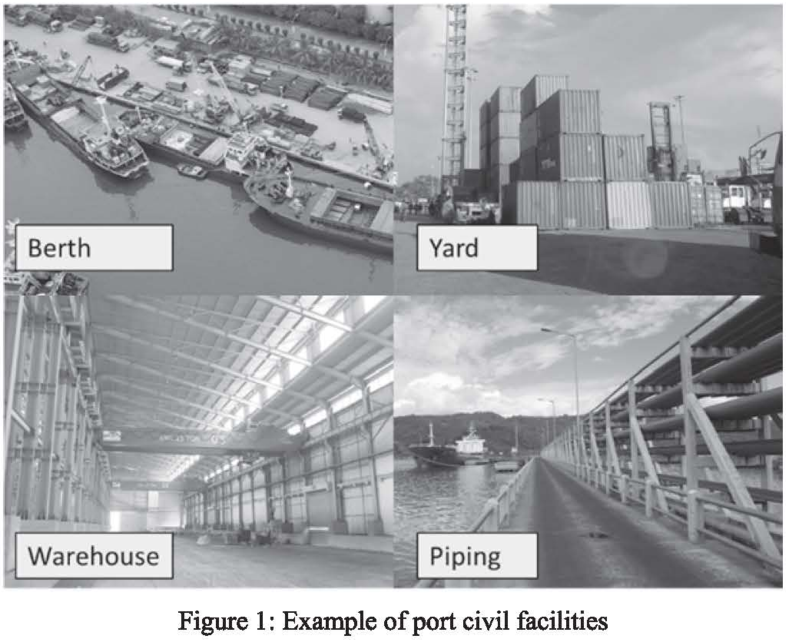

Figure 1: Example of port civil facilities

(c) *Warehouse*: The bulk storage buildings, warehouses, are usually constructed from concrete or timbers. They are facilitated with lightings, utility networks, marks, signs, and so on.

(d) Piping: The piping facilities (for receiving or dispatching liquid bulk) are usually equipped with lightings, electricity grids, marks, signs and so on.

Berths, yards, warehouse and piping structures with their equipment components need to be maintained regularly. The importance of port infrastructure maintenance can be viewed in the following three aspects (Zhang et al., 2017):

(a) Most port infrastructures need to be utilized for a very long period of time that structural deteriorations may always occur. For this reason, maintenance is a long-term work which aims to keep the lifetime risk below a target level.

(b) Port infrastructures are normally built along the coastal areas which demands frequent repair and retrofit works. As most port infrastructures are directly exposed to the harsh environment, a fast deterioration mechanism is usually expected in the constructed facilities. For example, the chloride-induced corrosion is a primary cause of most reinforced concrete structures in onshore marine environment. There is a need for effective and economic maintenance planning for these quickly-deteriorating structures.

(c) The consequences associated with port infrastructure failures due to insufficient maintenance can be enormous, not only causing a disruption at ports, but also a possible stoppage of the whole port supply chain. In other words, the condition of infrastructural performance affects the efficiency of port operations and associated sectors.

The port infrastructures as well as the equipment components attached to or existed on those infrastructures, which are significant in supporting the port operations, must be maintained.

2.2 ERP, Information Systems and ERP Frameworks

The Enterprise Resource Planning (ERP) system is a set of software that is used to organize, define, and standardize the business processes that are necessary to effectively plan and control an organization. Essentially, ERP applications are a computer model of organization business, embodying the products and processes, information flow, procedures, and relationships between functions and activities. Data analysis and reporting tools are also an essential part of the system, allowing full exploitation of the broad range of information that the system will manage. Another essential technology characteristic is the Internet connectivity. Modern systems employ thin-client, highly graphical and tailor able role-based user interface (APTEAN, 2017).

The computerized information system (IS) is an integrated set of people, processes, and mechanisms for collecting, storing, and processing data using the IT infrastructure to deliver an information toward a particular goal (Stair and Reynolds, 2014). Any specific IS aims to support the operations, management, and decision-making. ERP system is an IS, and vice versa. However, GIS, which is a system designed to capture, store, manipulate, analyze, manage, and present spatial or geographic data, is usually not a part of the ERP systems. To provide an easy and fast development of ERP systems, many vendors offer ERP frameworks. Each framework comes with a lot of modules (such as Accounting, CRM, HR, Marketing, MRP, and Sale) that can be used as is or customized by developers to meet organizations' requirements. One of the available frameworks is Odoo. Odoo is an ERP framework built upon a Model-View-Controller (MVC) architecture (Odoo, 2017). One of the primary goals of this architecture is to separate the visual display of the information from the business rules and management of the underlying data. With this architecture, the Model notifies the Controller of any data changes, and in turn, the Controller updates the data in the Views. The View can then notify the Controller of actions the user performed and the Controller will either update the Model if necessary or retrieve any requested data (Figure 2).

Figure 2: MVC architecture of Odoo ERP framework

Developers build an ERP system for specific organizations by developing Odoo module. An Odoo module is usually composed of data models, together with some initial data, views definitions (how data from specific data models should be displayed to the user), wizards (specialized screens to help the user for specific interactions), workflows definitions, and reports. The module contains the elements of: (a) Business objects: Declared as Python classes, these resources are automatically persisted by Odoo based on their configuration; (b) Data files: XML or CSV files declaring metadata (views or workflows), configuration data (modules parameterization), demonstration data and more; (c) Static web data: Images, CSS or JavaScript files used by the web interface or website.

Odoo clients can communicate with Odoo server using XML-RPC. As the logic of Odoo should entirely reside on the server, the client is conceptually very simple: it issues a request to the server and displays the result (such as a list of customers) in different manners (as forms, lists, calendars and so on). Upon user actions, it will send the modified data to the server. The data tier of Odoo is provided by a PostgreSQL relational database. While direct SQL queries can be executed from Odoo modules, most database access to the relational database is done through the Object-Relational Mapping (ORM). The data models are described in Python and Odoo creates the underlying database tables. All the benefits of RDBMS (unique constraints, relational integrity, efficient querying, and so on) are utilized when possible and completed by Python flexibility. The Odoo Model (as M in MVC) specifies how some data are structured, constrained, and manipulated. In practice, a model is written as a Python class that encapsulates: The fields composing the model, default values to be used when creating new records, constraints, and so on. It also holds the dynamic aspect of the data it controls: methods on the class can be written to implement any business need (for instance, what to do upon user action).

2.3 SVG and the use in GIS-ERP System

Scalable Vector Graphics or SVG is an XML-based graphics standard that enables Web documents to be smaller, faster, and more interactive (Eisenberg, 2002). Using SVG, an image is described as a series of geometric shapes. Rather than receiving a finished set of pixels, a vector viewing program receives the commands to draw shapes at specified sets of coordinates. Thus, vector graphics have one feature that makes them invaluable in many applications- they can be scaled without loss of image quality. There are many shapes that can be displayed, such as lines, circles, and polylines. Texts can also be added on a specific coordinate in the canvas using some choices of style, font, size, color and orientation. Object shapes can be grouped and an SVG may have many object groups. Later on, the viewing program can select which groups that will be displayed on the canvas/screen. Thus, in the context of GIS, a group of objects can be defined as a "layer". The GIS viewing program can be designed to display one or more layer of objects. Having those features and capabilities, SVG is a good option to display design or map of port facilities (wharves, yards, piping, and warehouses) as detailed images with its object labels or legends or information can be displayed in many scales on web browsers without losing its clarity. Also, objects in a specific facility can be grouped to form a particular layer. One problem that must be resolved: As the complete data of the facilities, including their attributes, will be stored in relational database tables, how to relate the tables with the SVGs? Fortunately, each group and shape object in SVG may have its own ID and name as depicted in a simple example of SVG with three groups of objects is as follows:

```
<svg>
 <g id="layer_BG" name="Background"> <rect
  width="800" height="1000" stroke="black"
  stroke-width="2" fill="cyan"/>
 </g>
 <g id="layer_1" name="Wharf">
   <circle class="layer_fac" cx="50" cy="50"
     r="16" stroke="black" stroke-width="2"
     fill="red" id="fac_Wharf-1"  name=
     "Wharf-1" />
   <circle class="layer_fac" cx="50" cy="120"
     r="16" stroke="black" stroke-width="2"
     fill="red" id="fac_Wharf_2" name=
     "Wharf-2"/>
 </g>
 <g id="layer_2" name="Yard">
  <circle class="layer_fac" cx="200" cy="50"
    r="16" stroke="black" stroke-width="2"
    fill="yellow" id="fac_Yard-1" name=
    "Yard-1" />
  <circle class="layer_facility" cx="300"
    cy="50" r="16" stroke="black" stroke-
    width="2" fill="yellow" id="fac_Yard-2"
    name="Yard-2" />
 </g>
</svg>
```

The ID and name of each SVG group and object can then be stored as table attributes, and the ID can be defined as the primary key of the corresponding tables. The SVG codes can also be stored in a table attribute.

3. Problems and Designed Proposed System

3.1 Problems

The problems of a big organization that operates cargo terminals with hundreds of structural facilities and its branches can be comprised as follows. The structural facilities should be maintained in order that they can operate smoothly at all times to support cargo loading and unloading. As the branches are spread across three islands, the headquarter administration needs to have the real-time reports (summary as well as detailed) of the current conditions of the facilities in order to perform proper maintenance actions promptly. Due to the large area of the ports, where each port has many structural facilities and each facility is of large dimensions, the management need to quickly and easily "pin point" the exact position of the facility as well as the locations that need attention (with damaged infrastructures) with its complete reports/performance regarding their conditions. They should also be able to record the appropriate repair actions into the system.

As depicted in (FAO, 2013), one of the purpose of GIS is facilities management. (Maia et al., 2017) have also proposed a WebGIS to maintain coastal structures, particularly breakwaters. They emphasize the importance of checking the structures on-site and recording the relevant data into WebGIS for the purpose of analysis and designing proper actions to avoid total damaged. Hence, web-based GIS is also proposed to be utilized by the headquarters as well as all the branches. Given the advantages of ERP framework (Subsection 2.2), the GIS will be developed as part of the ERP module.

3.2 Proposed Solution Concept

In essence, to resolve the organizations problems, we propose the following concept:

(1) The main structural facilities (wharves, yards, warehouses, and pipes) along with their most important components attached to them are the facilities that would be handled in the system. Weights are assigned for each facility and each component by administrator such that KPI for each facility as well as branch can be computed using a defined formula.

(2) Every component of facilities (for instance, for wharves: slabs, bollards and fenders; yards: floor and lightings) requires an appropriate procedure and specific periods of observation. Hence, we propose two types of observations, which are:

- (a) Monitoring: conducted monthly towards components that are easily damaged and need simple efforts to be observed (such as the components attached to the surface of facilities);
- (b) Inspection: performed annually or once every 4 years towards the components that are rarely damaged and require difficult efforts to be inspected (such as structures that are below the surface or submerged in the water);

(3) The results of monitoring and inspections, which are component status (ok/not-ok for monitoring, and good/ lightly-damaged/ heavily-damaged for inspection), descriptions, and photos taken at the site are input promptly into the system by field inspectors using a mobile app or by clicking objects on the facility maps. Then, the entered data is checked by a branch administrator. After branch managers reviewed the information of the damages, repair actions are also input into the system.

(4) Users can use the interactive maps of facility site plans to get the information (text-based as well as visual) of a particular facility and its components.

(5) Summary, performance (KPI computation results), and detailed reports (text-based as well as dash-board) of the current and historical conditions, and the repair action plans towards facilities can be accessed easily by executive users via web browser (to support the decision making).

The "traditional IS features" to record data into forms and generate reports will be implemented by customizing the modules provided by the selected ERP framework (see Subsection 2.2). Hence, the concept of the interactive map can be depicted below. At first, the map of a branch is displayed. Users can select whether they want to display all layers or just a layer. The map branch will include "mouse-over-able" and "click-able" objects of facilities. When an object is mouse-hovered, the information of the object will pop up. If a facility object is clicked, the map of the facility will appear and replace the map of the branch. Every facility object as well as the component of a facility will be displayed in color representing its performance (green: ok, red: damaged, grey: has not been observed). The branch as well as the facility page will also include its summary reports and KPI (Figure 3a). If the user is an inspector (who has the duty to input the monitoring and inspection data), once the map of a pop up (Figure 3b). The user can fill the form and save it. To facilitate the interactive maps, the system database store and manage the maps (SVG) of branches, which relates to their map of facilities. Each facility map must relate to its component layers, each component must relate to its check items layers (Figure 4). As described in Section 2.3, the ID of the SVG is used to relate between database tables.

3.3 System Architecture

Unlike the Web-based GIS discussed in (Afnarius et al., 2017) that was developed using PHP, Javascript and PostgreSQL/PostGIS, the proposed Marine Terminal Civil Facility Maintenance (MTCFM) is built using the ERP framework to take its advantages. The architecture is depicted on Figure 5.

It comprises of two main modules:

(1) Facility Maintenance ERP Module: used by several groups of users (inspector, branch administrator, branch manager, corporate division head and director). This module is rich with features that are customized from the libraries provided by the ERP frameworks as well as the ones designed with interactive map capability.

(2) Mobile App: used by facilities inspectors to download uninspected facilities, input, and upload the results of monitoring and inspections results. By using this app, inspectors have the advantage of taking facility pictures and inputting the component conditions using tablets that they carry while working in the remote fields.

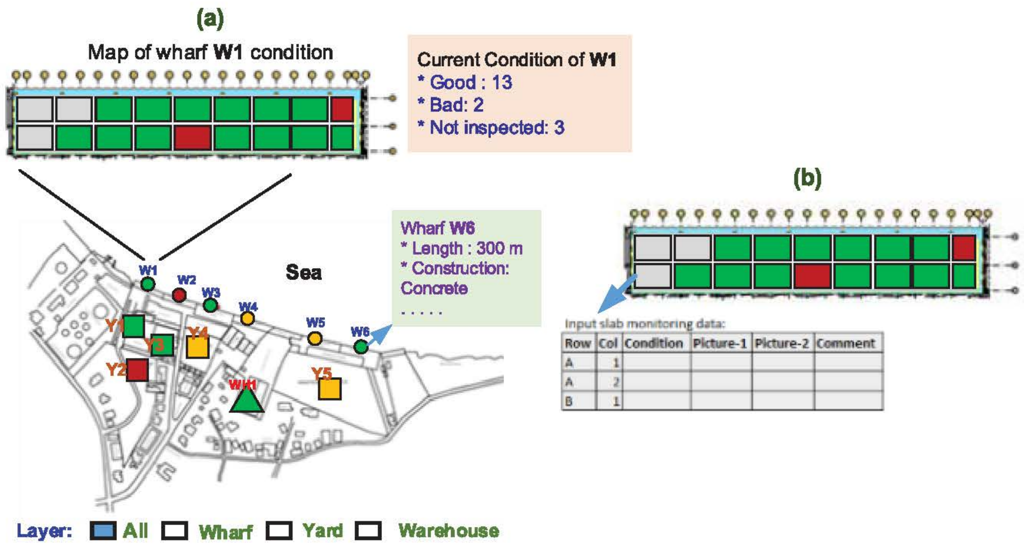

Figure 3: (a) Map of port civil facilities with its performance color; (b) Inspector click on a component layer map to pop up the form

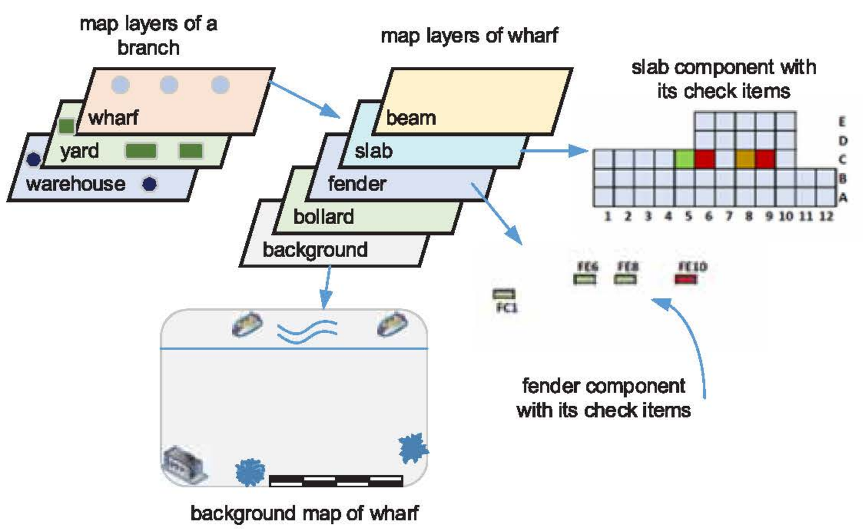

Figure 4: Map layers of branch, a wharf facility and component with its check item

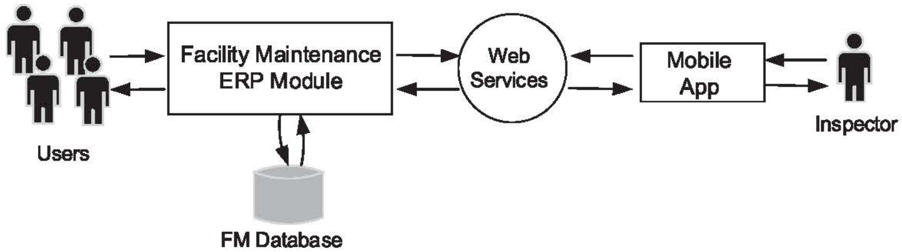

Figure 5: Facility maintenance system architecture

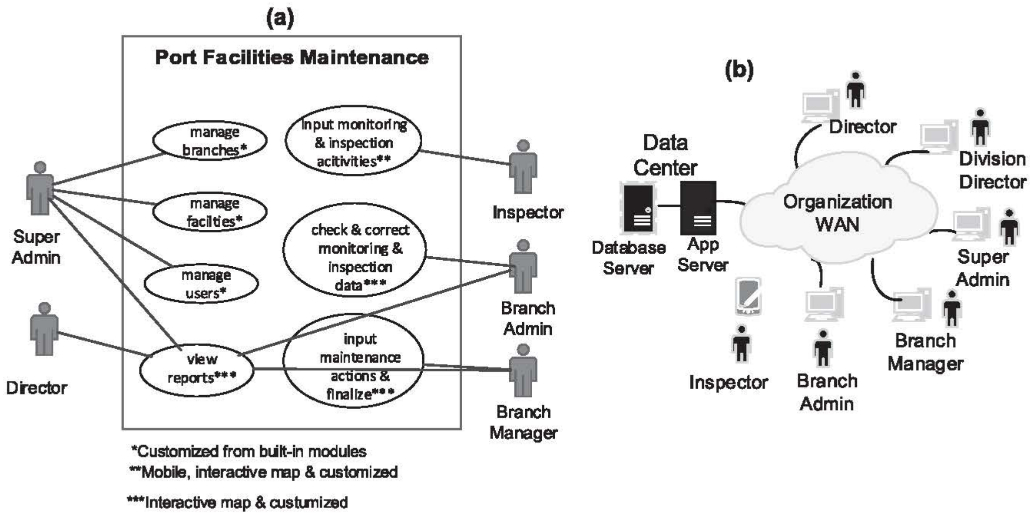

Figure 6: The use-case (a) and system accessed via WAN (b)

3.4 Use-Case

The use-case describing features for each group of users and features is depicted on Figure 6a. The color in the ellipsis indicate whether the features are provided by the module that are customized from built-in ERP framework, interactive maps as well as customized or mobile app. The users access the system via the organization Wide Area Network (WAN) using Internet browsers (Figure 6b). The reports that can be accessed by admin, manages and director include summary and detailed facilities conditions in every branch as well as branch and facility KPIs. As discussed in Subsection 2.2, an ERP module consists of data models, together with some initial data, views definitions (how data from specific data models should be displayed to the user), wizards (specialized screens to help the user for specific interactions) and reports. The users interact with the system via a client (such as web browser).

3.5 System Design

Based on the Odoo architecture, the proposed ERP system can be seen in Figure 7. The clients can be categorized into two modules:

(a) Customized from built-in client side Odoo module: It issues a request to the server and display the result (a list of facilities) in different manners (as forms, lists, and so on). Upon user actions (on the menus or forms), it will send modified data to the server.

(b) "Wizards" module that provides interactive maps: this client module is developed using CSS, Ajax, and JavaScript for HTML 5. Libraries for SVG map display are used as well. This wizard displays maps with layers that can be selected and has objects that can be mouse-hovered as well as clicked to get the object information or to pop-up forms (used to input data) if the user is the field inspector. This module also computes the facilities condition summary and KPIs, and displays the results in the dash-boards.

The server side module specifies how data are structured, constrained, and manipulated. It contains a number of models written as a Python class, having attributes, constructors and methods (the class diagram is depicted on Figure 8).

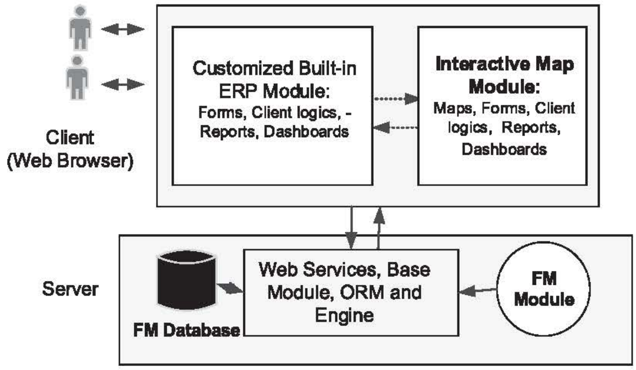

Figure 7: Facility maintenance (FM) modules

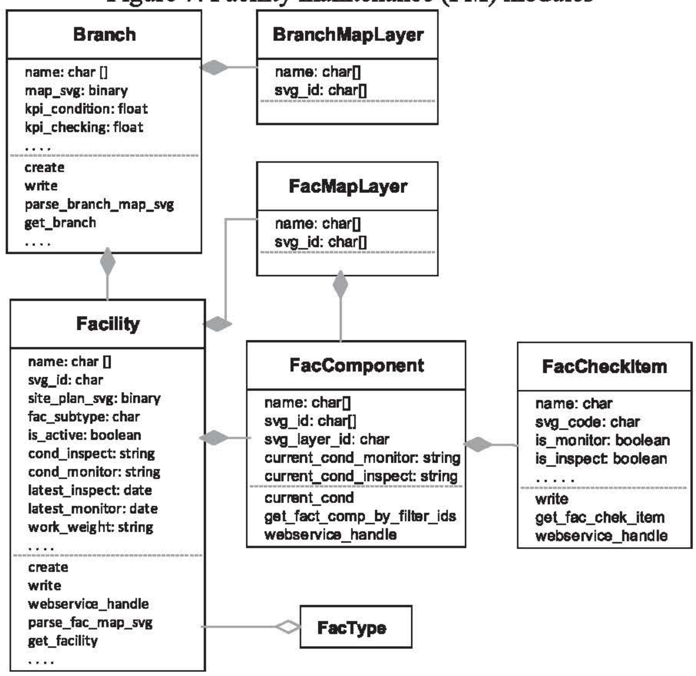

Figure 8: Class diagram related to maps data

Database Schema: The FM database schema that are created by the model can be comprised as follows:

Branch(IdBr, name, map_svg, …*other attributes*);
// the file of map_svg also contains branch port/terminal
layers (wharves, yards, warehouses, and piping that existed in the branch port/terminal area).
BranchMapLayer(Svg_Id, name, IdBr*(referring to Branch(IdBr)*);
Facility(IdFac, name, facility_plan_svg, IdBr*(referring to Branch(IdBr)*, …*other attributes*);
//the file of facility_plan_svg also contains component type layers (in the case of wharves: slabs, beams, fenders, and bollards) along with its components or object check items (in the case of wharf slabs, they are the element of slab construction).
FacMapLayer(Svg_Id, name);
FacComponent(IdFC, name, svg_layer_id, IdBr*(referring to Branch(IdBr)*, …*other attributes*);
FacCheckItem(IdFCI, name, svg_code, IdFC*(referring to FacComponent (*IdFC*)*, …*other attributes*);

Database tables and their attributes are conformed to the model classes, which is depicted on Figure 8. Other database tables for storing master data (facility type, users, and so on), facility weights (for computing KPI), many kinds of monitoring and inspection transactions, repair activities are also designed and implemented.

The Classes in the Model Module: The design of the main classes representing the branch, facilities and their component is shown in Figure 8. There are other classes that are not presented in this figure, such as classes that manage users, handle various transactions (inputting/updating data of facility monitoring and inspection activities, and facility status). Figure 8 shows that an object of branch has one or more objects of branch map layer, a branch map layer has one or more objects of facility, a facility has one or more objects of facility map layer, a facility map layer has one objects of facility component, an object of Facility also have one or more facility component, a facility component type consist of one or more components or objects of check Items (as illustrated in Figure 4). The brief description of classes and their main methods are as follows:

class Branch: Handle branches and their map of branch.
Methods:
create: Parse, check, clean the branch SVG file and save the related attributes to the corresponding records in Branch, BranchMapLayer and Facility.
write (update): Parse, check, clean the branch SVG file (if reloaded) and update the related attributes to the corresponding records in Branch, BranchMapLayer and Facility.
class Facility: Handle facilities and their maps.
Methods:
create: Parse, check, clean the facility SVG file and save the related attributes to the to the corresponding records in Facility, FacMapLayer, FacComponent and FacCheckItem.
write (update): Parse, check, clean the facility SVG (if reloaded), and update the related attributes to the corresponding records in Facility, FacMapLayer, FacComponent and FacCheckItem.
webservice_handle: Get records of a facility, its component types and components with all of their status, construct a complex object and return. The records can be filtered by list_Ids or user_Id.

The data files (see Subsection 2.2) containing XML files declaring metadata (views or workflows) and configuration data (modules parameterization) are declared accordingly such that they can be used by functions in accessing, computing and displaying the database content. Folder *static*: contains CSS describing the style of the HTML documents, JavaScript source code for client logic (computing report and dash-board material) and handling interactive maps as well as the non-map pages (by using styles defined in CSS), XML describing lots of HTML classes used in the CSS, and Lightbox JavaScript libraries (to manage images display).

The Navigation of Interactive Map: A menu "Monitoring & Inspection" (in FM ERP Module) is specifically provided for moving or navigating to pages (of the wizard module) with map interactivity capability. When that menu is clicked, a page with visual reports of every branch is displayed. Then, when a branch report is clicked, the branch map along with its filter widgets (to select year, facility type layer, condition, monitoring or inspection, report and dashboard for this branch) is displayed (Figure 9). Furthermore, if users hover on a facility object, the information of this facility will be popped up. Then, by clicking an object (a wharf, a yard, a warehouse, or a pipe) on the branch map, the corresponding facility with its filter widgets (to select the month, the year, the component type layer, the condition, the monitoring or inspection, the report for this facility) is displayed (Figure 10). When a user clicks on a component: if the user is a field inspector, a form for entering the monitoring/inspection results will be popped up; if other user, a report of monitoring/inspection for the corresponding component at the selected month-year, will be popped up.

Hence, there are two pages designed with interactive map, which are branch and facility page. These wizard pages are designed with: (a) JavaScript functions that display/render the map and all of the widgets (for filtering and to be clicked), and lots of on-hover, on-button/mouse-click, on-mouse-out, on-change of combo check-box, on-change of drop-down list functions, drawing objects in dash-board (such as monitoring and inspection charts, and monthly-time-series charts; (b) CSS definitions that describe the styles for those 2 pages; (c) XML definitions with many HTML classes used in the CSS.

4. Implementation

The designed have been implemented and the results are presented with sample of screenshots below:

(a) Figure 9: the window of a branch with its drop-down filter (month-year and condition) at the top left side and button filter (Monitoring and Inspection) at the top right, report dashboard (at the right side), menu and map with layer selection (at the left side). Zooming the map will display a clearer map, and mouse-over on a facility object displays its information.

(b) Figure 10: By clicking on a facility object on the map branch, then select a component (for instance a wharf), the facility component map with its filter (top), menu (left) and dashboard (right) are displayed (b). The results of monitoring/inspections of a facility item component can be inputted by clicking on a location on the map.

Figure 9: Map of a branch with its layers selection, dashboard and mouse over on a wharf

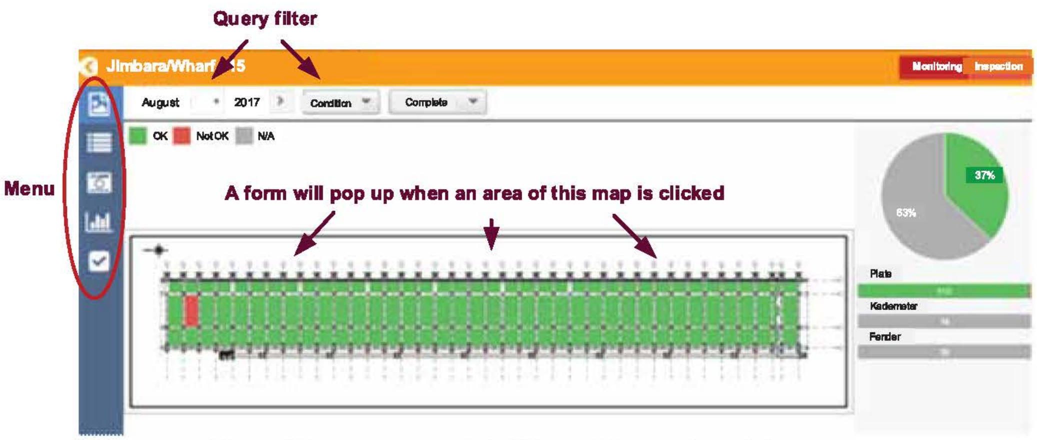

Figure 10: A component facility and its map-based form

5. System Evaluations

As it has been known that the most important parameter to measure the information system success is by conducting user acceptance test (UAT). There were eight sub-modules to be tested: (1) master data management; (2) monitoring data management using text-based forms; (3) inspection data management using text-based forms; (4) monitoring data management using map-based forms; (5) inspection data management using map-based forms; (6) user data management; (7) document data management; (8) monitoring and inspection data management using mobile app (used by field inspectors). The reports generation are part of sub-modules 2, 3, 4 and 6. A test plan for each sub-module was designed, which include list of features with their test scenario, data used, results, the tester person, and date of test. The excerpt of scenarios for each group of users (depicted on Figure 6) are as follows:

(a) Super Admin: to manage master data (users, branches, type of facilities, facilities and documents).

(b) Field Inspector: to record the results of facilities monitoring and inspection via web-based (using text forms and clicking map facilities) as well as mobile application (clicking map facilities after downloading the map).

(c) Branch Admin: to review monitoring and inspection results transaction data entered by field inspectors and marking the transaction (as completed or must be repeated).

(d) Branch Manager: to review the completed monitoring and inspection results transaction data, to record maintenance actions as well as to review reports and dash-boards.

(e) Director: to read/review maps of branch and facilities conditions, text reports, KPI and graphical reports (in the dash-boards).

The user acceptance test (UAT) was conducted at the end of the development, on the early of August 2017. The participants were users from the largest and busiest marine terminals Indonesia operating in west of Java, south of Sumatera and west of Kalimantan island. The excerpts of UAT results is as follows:

(a) Sub-module 1, 2, 3, 6, 7: All functions related to data recording and reports viewing are accepted. The part where the users experiencing difficulties are viewing information of a facility by clicking an object in the map. When the object is small, it is hard to "pin-point" the exact location. Thus, the map must be zoomed such that facility objects are easily clicked.

(b) Sub-module 4, 5: Compared to the text-based form, the map-based forms are easier to use but slower as users need to wait the map displayed on the screen. But users accept the functions.

(c) Sub-module 8: The speed of downloading maps from the server into the mobile device and uploading pictures into the server depends on the map and picture size as well as the Internet connection. To avoid waiting too long, users need to download and upload data partially. The functions are accepted with request for faster maps download.

The problem of (b, c), which is slow in downloading maps, has been addressed by "optimizing" the size of SVG files. Objects that are unnecessary are removed from files to reduce their size. By doing this, most files' size can be reduced by 1/3 to ½ of their original size. After this, the downloading speed has been accepted. At the end of August 2017, users trainings were conducted and attended by users from 12 port branches. The system was launched on the beginning of October 2017, and since then it has been used by all users from all branches via the corporate WAN. Based on the above facts, we conclude that the proposed system has been successful.

6. Conclusion and Further Works

Web-based ERP system with GIS module can be designed and implemented to resolve the problems in marine cargo terminals that are related to facilities maintenance. By using the system, field inspectors who gather the data of the conditions of facilities can input the data via "traditional menu and forms" as well as by easily clicking facility maps. The branch management can record repair actions into the system. The executive users can view the summaries and detailed reports as well as branch and facility KPIs by easily clicking branch and facility maps or using "traditional menu" (text-based). An ERP module can be extended such that it has a GIS sub-module. Users have the advantages of using interactive maps and rich functions provided by ERP frameworks. Developers gain the advantage of faster system development. They are able to customize the available ERP functions quickly while also developing the GIS module that takes more time and resources. For further works, the system will be enhanced by integrating it with Google Maps. Each facility and its components will be accessed by clicking a "mark" on this map. Other feature that will be added in the mobile app is recording the monitoring and inspection results on site and based on the GPS coordinates of the mobile device.

Acknowledgements

We would like to thank PT Pelabuhan Indonesia II for funding this project.

References

Afnarius, S., Putra A. T., Tamara, A., Dinata, U. G. S., Ichwana, D. and Akbar, F., 2017, Web GIS Development for Minangkabau Customary Village: A Case Study of Padang Lua Village, West Sumatra, Indonesia. *International Journal of Geoinformatics*, Vol. 13(4), 1-12.

Alderton, P. M., 2008. *Port Management and Operations*, Informa. London.

APTEAN, 2017, *ERP System Definition*, http://www.aptean.com/additional-crm-and-erp-related-links-pages/erp-resources-folder/erp-system-definition (accessed 27 August 2017).

Eisenberg, J. D., 2002. *SVG Essentials*, O'Reilly Publ., USA, 1st ed.

Food and Agriculture Organization (FAO) of the United Nations, 2013. *Advances in Geographic Information Systems and Remote Sensing for Fisheries and Aquaculture*. Rome, Italy.

International Navigation Association (INA), 1998. *Life Cycle Management of Port Structures General Principles*. Report of Working Group No. 31 of the Permanent Technical Committee II. PIANC General Secretariat, Brussels, Belgium.

Kia, M., Shayan, E. and Ghotb, F., 2000, The Importance of Information Technology in Port Terminal Operations, *International Journal of Physical Distribution & Logistics Management*, Vol. 30 No. 3/4, 2000, 331-344.

Maia, A., Rodrigues, A., Lemos, R., Capitão, R. and Fortes, C., 2017, A Web Platform for the Systematic Monitoring of Coastal Structures. *In*

Proc. of the 3rd International Conference on Geographical Information Systems Theory, Applications and Management (GISTAM 2017), 102-111.

Odoo, 2017, *Basic of Odoo Architecture*, http://www.odoo.gotodoo.com/basics-odoo-architecture/ [accessed 30 July 2017]

Odoo, 2017, *Building a Module*, https://www.odoo.com/documentation/10.0/howtos/backend.html [accessed 13 August 2017].

Petering, M. E. H., 2011, *Introduction to the Container Shipping Industry*, Department of Industrial and Manufacturing Engineering. University of Wisconsin – Milwaukee.

Stair, R. and Reynolds, G., 2014, *Fundamentals of Information Systems*, Course Technology. USA.

Zhang, Y., Kim, C. W., Tee, K. F. and Lam, J. S. L., 2017, Optimal Sustainable Life Cycle Maintenance Strategies for Port Infrastructures, *Journal of Cleaner Production*, Vol. 142, 1693-1709.