

**Part A.**

The results of this activity should be a list of system requirements also known as a system requirements specification (SyRS) for the proposed system.

System requirements for the proposed system are separated into three categories: functional, operational, and maintenance. The functional requirements of a system are requirements that define what the system is supposed to do. The operational requirements of a system are the requirements that define how the system is supposed to operate usually in terms of performance. The maintenance and support requirements of a system are requirements that focus on the reliability, maintainability, and supportability of the system.

**Deliverable:**

Provide the functional, operational, and maintenance requirements for *one of your subsystems* from your proposed system.

An example of these requirements is shown below. Note that these are the requirements for three different subsystems.

**a) System Requirements for the Forecasting Subsystem**

A portion of the functional requirements for the Wait Time Forecasting subsystem are as follows:

- The subsystem shall create customer wait time forecasts for full-service restaurants located in the continental United States.
- The customer wait time forecasts shall have an average daily mean absolute error value of five minutes or less for each restaurant location.
- The subsystem shall only use data collected from the Internet as predictor variables.
- The subsystem shall generate customer wait time forecasts for each restaurant in fifteen-minute increments.
- The subsystem shall be able to integrate with the back end subsystem.
- The subsystem shall provide customer wait time forecasts to the back end subsystem.

- The subsystem shall generate customer wait time forecasts during the normal operating hours for the restaurant.
- The subsystem shall generate customer wait time forecasts for each restaurant one week in advance.
- The subsystem shall not require the purchase or installation of additional hardware components at any restaurant.
- The subsystem shall allow users to remotely access, maintain, and update it.

The operational requirements for the proposed subsystem are as follows:

- The subsystem should not cost more than five thousand U.S. dollars to develop.
- The subsystem should not cost more than ten thousand U.S. dollars per year to operate.

The subsystem maintenance and support requirements for the proposed subsystem are as follows:

- The subsystem shall have an annual uptime of ninety-nine-point nine percent. (This allows for only 8.76 hours of downtime per year.)
- Each component of the subsystem shall automatically check for any necessary software updates on a daily basis.
- The subsystem shall automatically download and install any required software updates.
- The subsystem shall not cost more than \$1,500 U.S. dollars to dispose.

#### **b) System Requirements for the Back End Subsystem**

A sample of the functional requirements for the back end subsystem are as follows:

- The subsystem shall have a database to store customer and restaurant information.
- The subsystem shall store a profile of the customer consisting of a photo, name, email address, mobile number, and password.
- The subsystem shall allow customers to edit their customer information.

- The subsystem shall determine and provide continuous walking directions to a restaurant from the user's current location.
- The subsystem shall determine and provide continuous driving directions to a restaurant from the user's current location.
- The subsystem shall provide a method for payments from customers to be transferred to the restaurant within a 24-hour period using a list of payment options.
- The subsystem shall have a restaurant review feature consisting of one to five stars where one star is low and five stars is high.
- The subsystem shall allow each customer who has paid his bill to enter an optional narrative review of the restaurant as well.
- The subsystem shall allow the owners of the system to generate reports detailing system activity. The subsystem shall include the following reports: location fees, location sales, customer history, and restaurant reviews.
- The subsystem shall require username and passwords for each administrative user.
- The subsystem shall have a web browser interface which allows the owners of the system to administer the system.
- The subsystem shall allow a restaurant employee to update the subsystem to indicate that the customer has vacated the restaurant.
- The subsystem shall capture the receipt data and the amount owed by the customer at the conclusion of the meal.
- The subsystem shall have a web interface accessible over a modern web browser.
- The subsystem shall provide 100% paperless reporting as an option.
- The subsystem shall allow users to remotely access, maintain, and update it.

A sample of the operational requirements for the back end subsystem are as follows:

- The subsystem should not cost more than twenty-five thousand U.S. dollars to develop.
- The subsystem should not cost more than one hundred thousand U.S. dollars per year to operate.
- The subsystem shall not cost each restaurant more than two thousand U.S. dollars per year to operate.
- The subsystem should not cost the restaurant more than two hundred (\$200) U.S. dollars to implement.

The subsystem maintenance and support requirements for the proposed subsystem are as follows:

- The subsystem shall have an annual uptime of ninety-nine-point nine percent. (This allows for only 8.76 hours of downtime per year.)
- Each component of the subsystem shall automatically check for software updates on a daily basis.
- The subsystem shall automatically download and install any required software updates.
- The corporate office shall maintain an inventory level of physical components to replace failures as needed.
- Failed components shall be replaced from the corporate office within two business days of a notification of a failure.
- The physical components of the subsystem shall be returned from the corporate office to the vendor(s) for replacement or to a licensed repair service for repair as appropriate.
- The subsystem shall not cost more than \$1,500 U.S. dollars to dispose.

#### **c) System Requirements for the Mobile Application Subsystem**

Several examples of the functional requirements for the Mobile Application subsystem are as follows. Again, complete system requirements are found in Appendix C.

- The subsystem shall have an app which allows customers to install and use the subsystem over their own smart devices.
- The subsystem shall allow users to see restaurants on a map in a one mile radius by default.
- The subsystem shall allow users to be able to sort through available restaurants based on best match, location, distance, price, category, and wait time.
- The subsystem shall allow customers to view reviews of restaurants provided by other customers.
- The subsystem shall provide a copy of the receipt to the customer which corresponds to the entire purchase price for the meal.
- The subsystem shall allow a customer to pay for his meal through the app via a credit card or debit card transaction.
- The subsystem shall provide a customer with a history of restaurant receipts.
- The subsystem shall have a default tip amount of twenty percent that is added automatically to each bill.
- The subsystem shall allow customers to adjust their tip amount for one hour after departing the restaurant.
- The subsystem shall allow only customers who have paid their bill to leave a review of the restaurant.
- The subsystem shall require username and passwords for each user.
- Customer users shall be allowed to use their Facebook profile as a login.
- The subsystem shall be capable of properly functioning on Android devices running

version 7.0 or later.

- The subsystem shall be capable of properly functioning on iOS devices running iOS version 7.0 or later.
- The subsystem shall provide 100% paperless reporting as an option.
- The subsystem shall allow users to remotely access, maintain, and update it.

The operational requirements for the proposed subsystem are as follows:

- The subsystem should not cost more than fifty thousand U.S. dollars to develop.
- The subsystem should not cost more than one hundred thousand U.S. dollars per year to operate.

The subsystem maintenance and support requirements for the proposed subsystem are as follows:

- The subsystem shall have an annual uptime of ninety-nine-point nine percent. (This allows for only 8.76 hours of downtime per year.)
- Each component of the subsystem shall automatically check for software updates on a daily basis.
- The subsystem shall automatically download and install any required software updates.
- The subsystem shall not cost more than \$2,000 U.S. dollars to dispose.

## **Part B.**

Take a set of requirements and translate them into Use Cases. You will find it useful to refer to your answers to Part A above. You should use the requirements to develop use cases (instead of requirements lists) to express those requirements.

All of the steps below can be done using Microsoft Word, Excel, or Visio which are readily available on campus. Steps 3, 4, 5, 7, and 8 must be done in Visio for a much better appearance.

Do the following:

1. Identification of Use Cases: Provide a name and a one-sentence description of four or more use cases that describe the functioning of the system. (See figure 3-9 in the

textbook) To determine the use cases and actors, it may be helpful to ask some of the following questions:

- Who uses the system?
  - Who installs the system?
  - Who starts up the system?
  - Who maintains the system?
  - What other systems interface with this system?
  - Who provides information to the system?
  - Who extracts information from the system?
  - Does this system do anything automatically at preset times?
2. Determine the actors. (any outside entities: people, systems, etc.) that interact with your system. Note the following: Actors are not necessarily people. They can be physical devices, other software systems, etc. (See figures 3-10a – 3-10c in the textbook.)
  3. Use case diagram: Draw a use case diagram that includes these use cases and the actors you have specified. Arrows are used indicate an initiator of activity. They point from primary actors to the use case, and they point from the use case to its secondary actors. (See figure 3-12 in the textbook.)
  4. Domain model class diagram: Draw a subsystem domain model class diagram for one of your subsystems. (See figure 4-24 in the textbook.)
  5. State machine diagram: Draw a state machine diagram for a class in your system. (See figures 4-30 and 4-32 in the textbook.)
  6. Use case description: Create a fully developed use case description for a use case. (See figure 5-2 in the textbook.)
  7. Activity diagram: Create an activity diagram for a use case. (See figures 5-4 – 5-6 in the textbook.)
  8. System sequence diagram: Create a system sequence diagram for a use case. (See figures 5-7 – 5-11 in the textbook.)
  9. CRUD: Create a CRUD matrix for at least five use cases and their corresponding domain classes. It is okay if some of the C, R, U, and D entries in the matrix are blank. (See figure 5-13 in the textbook.)

### **Deliverable:**

For parts A and B of this assignment, submit a single .doc, .docx, or pdf file to the appropriate assignment drop box in the class Canvas site. Your work from Word, Excel, and Visio for all of the steps above should be combined into a single Word document with everything clearly labeled

and neatly organized. Only one group member should submit the assignment. If multiple group members submit, then your score for the entire group will be penalized.