**REQUIREMENTS ANALYSIS DOCUMENT**

1. **Introduction**
   1. **Purpose of the System**

There are so many social media websites and most of these are frequently used. However, top social media websites don’t focus on a topic. There is no social media website about culinary art concept.

Specifically, our project satisfies the need. It provides an environment to share people’s idea or an information about gastronomy or culinary. In other words, our project creates a free platform for discussion and forming an opinion.

* 1. **Scope of the System**
  2. **Objectives and Success Criteria of the Project**

Finishing the project, how do you decide whether or not it is successfully completed?

* 1. **Definitions, Acronyms, and Abbreviations**

This subsection should provide the definitions of all terms, acronyms, and abbreviations required to properly interpret the RAD.

* 1. **Overview**

This subsection should:

* Describe what the rest of the RAD contains
* Explain how the RAD is organized.

1. **Current System**

If the new system **will replace an existing system**, this section describes the functionality and the problems of the current system. Otherwise, remove this part.

1. **Proposed System**

Documents the requirements elicitation and the analysis model of the new system

* 1. **Overview**

Presents a functional overview of the system.

* 1. **Functional Requirements**

Describes the high-level functionality of the system.

* 1. **Nonfunctional Requirements**

**This part is extremely important. Please refer to SE301 (SOFT3101) presentation to clearly understand what to write in this subsection. The presentation is also added to the COMP4902 page.**

Describes user-level requirements those are not directly related to functionality. This includes usability, reliability, performance, supportability, implementation, and interface, operational, packaging, and legal requirements.

* + 1. **Usability**
    2. **Reliability**
    3. **Performance**
    4. **Supportability**
    5. **Implementation**
    6. **Interface**
    7. **Packaging**
    8. **Legal**
  1. **System Models**

Describe the scenarios, use cases, object model, and dynamic models for the system. This section contains the complete functional specification, including mock-ups illustrating the user interface of the system and navigational paths representing the sequence of screens.

* + 1. **Scenarios**

A scenario is an instance of a use case.

* + 1. **Use case model**

A use case is a generalization of a number of scenarios. Therefore, the number of scenarios must be equal to or greater than the number of use cases.

* + 1. **Object model**

The analysis object model, depicted with UML class diagrams, includes classes, attributes, and operations. The analysis object model is a visual dictionary of the main concepts visible to the user.

* + 1. **Dynamic model**

The dynamic model is depicted with sequence diagrams and with state machines. Sequence diagrams represent the interactions among a set of objects during a single use case. State machines represent the behavior of a single object (or a group of very tightly coupled objects). The dynamic model serves to assign responsibilities to individual classes and, in the process, to identify new classes, associations, and attributes to be added to the analysis object model.

When working with either the analysis object model or the dynamic model, it is essential to remember that these models **represent user-level concepts, not actual software classes or components.**

* + 1. **User interface—navigational paths and screen mock-ups**

1. **Glossary**

To establish a clear terminology, developers identify the **participating objects** for each use case. Developers should identify, name, and describe them unambiguously and collate them into a glossary.

1. **References**

This subsection should:

* Provide a complete list of all documents referenced elsewhere in the RAD, or in a separate, specified document.
* Identify each document by title, report number - if applicable - date, and publishing organization.
* Specify the sources from which the references can be obtained.

The following is an example of listing a book in this section. Check the text to see how it is cross referenced (The whole document is based on [1]).

1. Bruegge B. & Dutoit A.H.. (2010). *Object-Oriented Software Engineering Using UML, Patterns, and Java*, Prentice Hall, 3rd ed.