**COMP 3059 – Capstone Project I**

**Software Requirements Analysis and Design Assignment**

This assignment is an overview to gather the software needs with requirements analysis and help to proceed with the design.

The requirements analysis helps to break down functional and non-functional requirements to a basic design view to provide a clear system development process framework. It involves various entities, including business, stakeholders and technology requirements.

The design is the activity following requirements specification and before programming. Software design usually involves problem solving and planning a software solution.

To work on this assignment you could use the references and a sample template given below. The sample template can be customised to suit the nature of your project.

Reference Readings/Example:

<http://www.uacg.bg/filebank/acadstaff/userfiles/publ_bg_397_SDP_activities_and_steps.pdf>

<http://gmitweb.gmit.ie/pdunne/sweng/03-Requirements.pdf>

[www.cse.msu.edu/~chengb/RE-491/Papers/SRSExample-webapp.doc](http://www.cse.msu.edu/~chengb/RE-491/Papers/SRSExample-webapp.doc)

<https://nces.ed.gov/pubs2005/tech_suite/part_2.asp>

Reference template:

[www.tricity.wsu.edu/~mckinnon/cpts322/cpts322-srs-v1.doc](http://www.tricity.wsu.edu/~mckinnon/cpts322/cpts322-srs-v1.doc)

# 1.0 Introduction

The Nite Bite’s application basic system requirement is to allow customers to interact with vendors(restaurants and grocers) in order to purchase the food. Transaction system, database interaction, rating system, and geolocation api are the main requirements of the system

## Purpose

The purpose of the system is to allow customers to find affordable food nearby using the google maps api, and querying the vendor database, that is updated regularly by the vendor. The rating system should be fairly detailed to instill confidence in purchase. The transaction system allows for safe purchase by customer, and direct pay to vendor.

## Scope

The users of the Nite Bite system will be customers, vendors, and system management. The customers are querying the vendors database for food in their location, while the vendor’s are uploading daily or hourly food stocks. The interaction between vendor and database should be fairly easy, because of the frequency of interaction. The rating system should be easy for the customer to use, and reliable, to instill confidence in purchase.

Transaction system must be safe. These are the main objectives of the system, on a macro level.

# System Overview

The System Overview section introduces the system context and design.

## Project Perspective

The Project Perspective describes the context and origin of the system by defining whether the system is:

* a follow-on member of a system family
* a replacement for existing systems, or
* a new self-contained system.

## System Context

The System Context describes the resulting software within the business case, including strategic issues in which the system is involved or which it specifically addresses.

## General Constraints

General Constraints identify any business or system constraints that will impact the manner in which the software is to be:

* specified
* designed
* implemented, or
* tested.

## Assumptions and Dependencies

Assumptions:

-Team members will be available for all important production meetings and development stages

-Version control will be organized

-System will be dependable and safe

-Changes to system during testing period should be seamless

-Database design should be straightforward and simple to use for vendors

## 3.0 Functional Requirements

-The costumer uses a log in function/or registration if new to system.

-The customer uses a search function, for potential food pickup in nearby area, using google maps API

-The vendor posts database updates regarding food inventory/food available.

-The customer uses a payment system to purchase food that is available

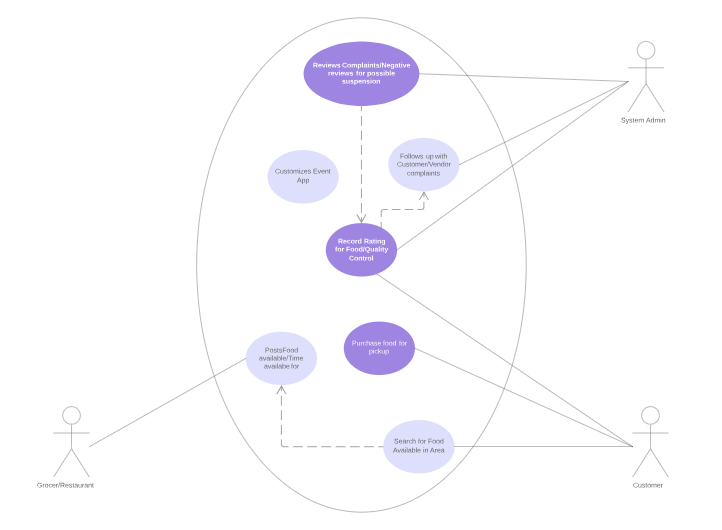
### 3.1 <Functional Requirement or Feature #1>

* Introduction
* Inputs
* Processing
* Outputs

...

## 3.2 Use Cases

### 3.2.1 Use Case #1



**3.3 Data Modelling and Analysis**

* Normalized Data Model Diagram
* Activity Diagrams
* Sequence Diagrams
* UML Class Diagram

**3.4 Process Modelling**

* Data Flow Diagram

## 4.0 Non-Functional Requirements

The non-functional requirements for a system are typically constraints on the functional requirements – that is, not what the system does, but how it does it (e.g. how quickly, how efficiently, how easily from the user’s perspective, etc.).

### Non-functional requirements may exist for any of the following attributes – Performance, Reliability, Availability, Security, Maintainability, Portability.

Often these requirements must be achieved at a system-wide level rather than at a unit level. State the requirements in the following sections in measurable terms (e.g., 95% of transaction shall be processed in less than a second, system downtime may not exceed 1 minute per day, etc).

## 5.0 Logical Database Requirements

Will a database be used? If so, what logical requirements exist for data formats, storage capabilities, data retention, data integrity, etc?

## 

## 6.0 Other Requirements

Additional requirements, if any.

**7.0 Approval**

The signatures below indicate their approval of the contents of this document.

|  |  |  |  |
| --- | --- | --- | --- |
| Project Role | Name | Signature | Date |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |