|  |  |
| --- | --- |
| **School of Computing & Digital Media** |  |
|  |  |

**Group 1 Assignment Report:**

UK Building Maintenance Services

**Student Names:** Gheorghe Mitrea (17014657), Patrick Shaw (17026810), Andrei Cirimpei (17006966), Abhik Modi (17026032)

**Course:** BEng (Hons) Software Engineering (Top-up)

**Supervisor:** **Dr Tingai Wang**

**January 2018**

Table of Contents

[Abstract 3](#_Toc505631880)

[Introduction 4](#_Toc505631881)

[1. System architecture design 5](#_Toc505631882)

[1.1 Entity-Relationship Diagram 6](#_Toc505631883)

[1.2 Functionality Needs to be Handled Using Use Case Diagram 6](#_Toc505631884)

[1.3 UML Class Diagram 8](#_Toc505631885)

[1.3.1 Class Diagram Inheritance 10](#_Toc505631886)

[2. System/Software Requirements- 10](#_Toc505631887)

[2.1 Functional Requirements 11](#_Toc505631888)

[2.2 Non-Functional Requirements 11](#_Toc505631889)

[2.3 User Requirements 12](#_Toc505631890)

[3. Software design and its specification 13](#_Toc505631891)

[References 16](#_Toc505631892)

List of Figures

[Figure 1 Entity Relationship diagram 6](file:///D:\ADVANCED%20SOFTWARE%20ENGINEERING\Group%201%20Report-part%201.docx#_Toc505631893)

[Figure 2 UML-Use Case Diagram 7](file:///D:\ADVANCED%20SOFTWARE%20ENGINEERING\Group%201%20Report-part%201.docx#_Toc505631894)

[Figure 3 UML class diagram 9](file:///D:\ADVANCED%20SOFTWARE%20ENGINEERING\Group%201%20Report-part%201.docx#_Toc505631895)

[Figure 4 Class diagram for inherited classes 10](file:///D:\ADVANCED%20SOFTWARE%20ENGINEERING\Group%201%20Report-part%201.docx#_Toc505631896)

[Figure 5 Deployment Diagram 13](file:///D:\ADVANCED%20SOFTWARE%20ENGINEERING\Group%201%20Report-part%201.docx#_Toc505631897)

[Figure 6 Login Page 14](file:///D:\ADVANCED%20SOFTWARE%20ENGINEERING\Group%201%20Report-part%201.docx#_Toc505631898)

[Figure 7 Successfully login 14](file:///D:\ADVANCED%20SOFTWARE%20ENGINEERING\Group%201%20Report-part%201.docx#_Toc505631899)

[Figure 8 Database view to add category 15](file:///D:\ADVANCED%20SOFTWARE%20ENGINEERING\Group%201%20Report-part%201.docx#_Toc505631900)

[Figure 9 Customer details saved in database 15](file:///D:\ADVANCED%20SOFTWARE%20ENGINEERING\Group%201%20Report-part%201.docx#_Toc505631901)

[Table 1 User Requirements 12](#_Toc505631909)

# Abstract

This report provides a brief description of an approach that will be used to design a software application. Advanced software engineering involves the use of different software tools to design a program based on different software platforms. Programming methods like SQL, Xampp server can be used to design the code for the application. Design and implementation of an online customer service system are done on the basis of requirements collected from the users. System development lifecycle model will be used for system design. Requirement specification is usually done by using a suitable fact-finding technique. This enables the system designers to include all the necessary requirements of users. The system will be useful for the organisation to provide repair and maintenance service to the customers. This app will enable the customers to register/login and can access the services provided by the company. This report discusses software development process for an enterprise.

# Introduction

The architecture design of the system is defined as an arrangement of different interface components into one or more structures that includes functional requirements. The role of each component is vital in software engineering process of system’s development. The different phases of systems development lifecycle are used to make the application reliable to use. For this, skilled developers are needed to make an effective use of the interface components. The user interface design has to appealing and usable to make the system better in performance. Developers take notice of the current needs of customers and which can suit best for the system designing. Front end and back end processes are designed according to the functionality.

According to the scenario, an application is required by an enterprise to manage the customer services. The company, ‘UK Building Maintenance Services’ deal with repairing and provide maintenance for home services such as painting, carpentering, roofing, plumbing etc. The company needs a desktop application that can be accessed by the customers as well as the staff members such as the Admins and the engineers. The system should have specific features as per the requirements of users which will be described in the next sections.

# System architecture design

The system architecture is responsible for designing, modifying, evaluating and enforcing user requirements with the help of software development tool. Software designing is divided into different phases like lexical, syntactic and semantic. System architecture consists of elements that are software and hardware. A good architecture comprises an algorithm that represents the variables, functions, and methods that have to be used in the program.

## Entity-Relationship Diagram

The diagram below represents the entities, their attributes, and relationship between them. These entities are used to design the database system for the proposed desktop application. The entities are customer, job, feedback, login, engineer, and category. This diagram represents the physical design of the proposed system.

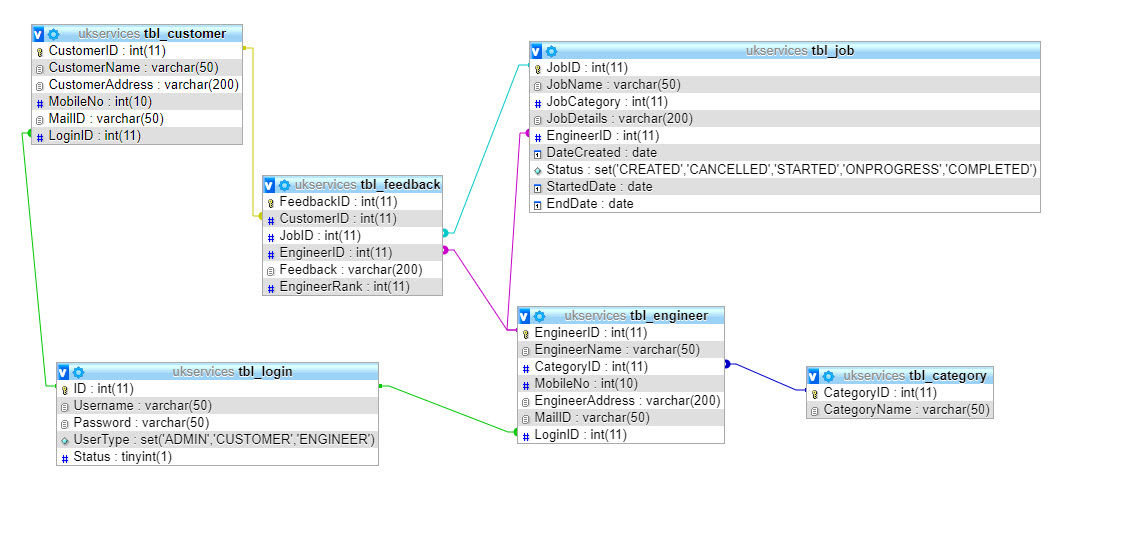


Figure 1 Entity Relationship diagram

## Functionality Needs to be Handled Using Use Case Diagram

* Login Page -- Login or Register
* Registration Page - Register as Customer or Engineer
* Login Page for Customer - 1. Create Job 2. Cancel job, 3. Feedback
* Login Page for Engineer - 1. Select a job 2. Update the job status (start, completion and close)
* Login Page for Admin - 1. Approve Engineer, 2. Assign an Engineer for Job, 3. View Report, 4. De Register an Engineer

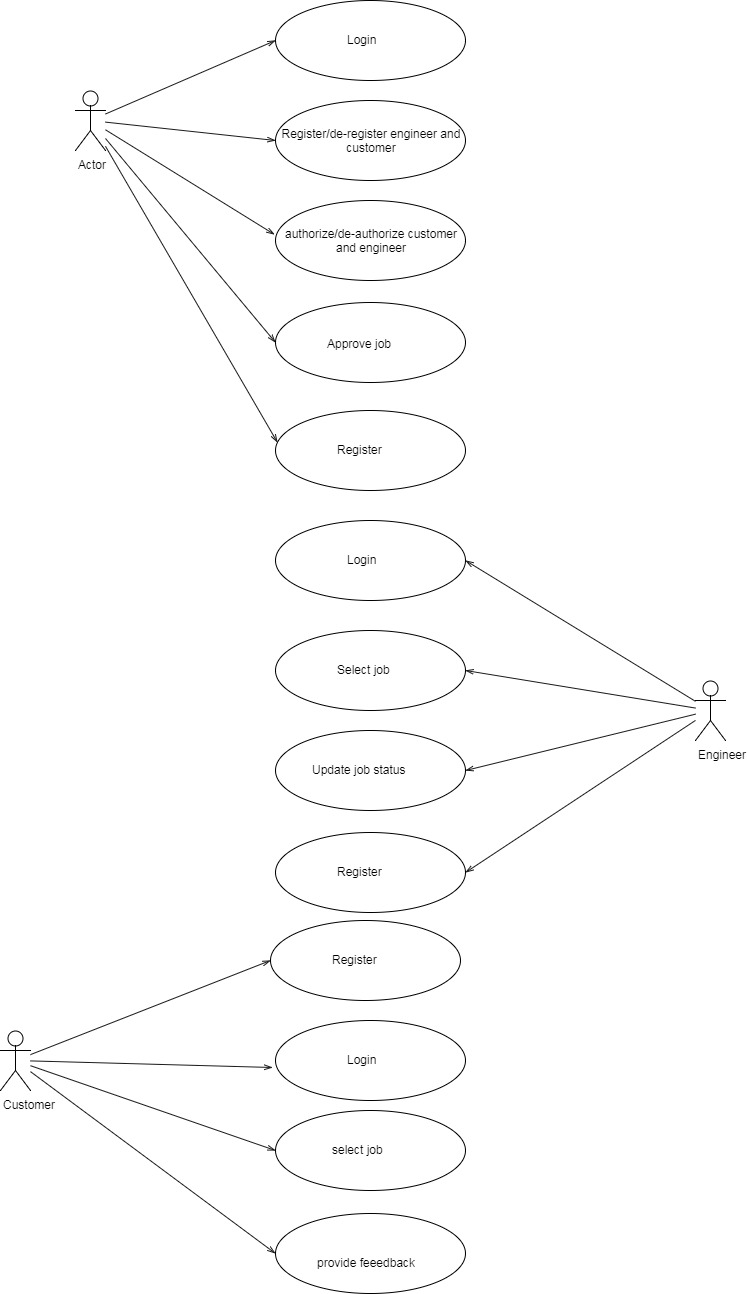


Figure 2 UML-Use Case Diagram

The above diagram represents use case diagram of the unified modeling language. A use case diagram is used to represent the functionality of a system. An actor is defined who performs the different operations in the system. The functionality of a system is represented by an ellipse. The arrows determine which process is performed by an actor. In this diagram, there are three different actors used.

## UML Class Diagram

This type of diagram is used to represent the static view of the proposed system. Class diagram represents the attributes and the operations performed by a class. These diagrams are used for the analysis and design of the structure for the proposed system. Class diagram is beneficial as it determines the component that is essential for the designing of the system (Tutorials Point, 2017).

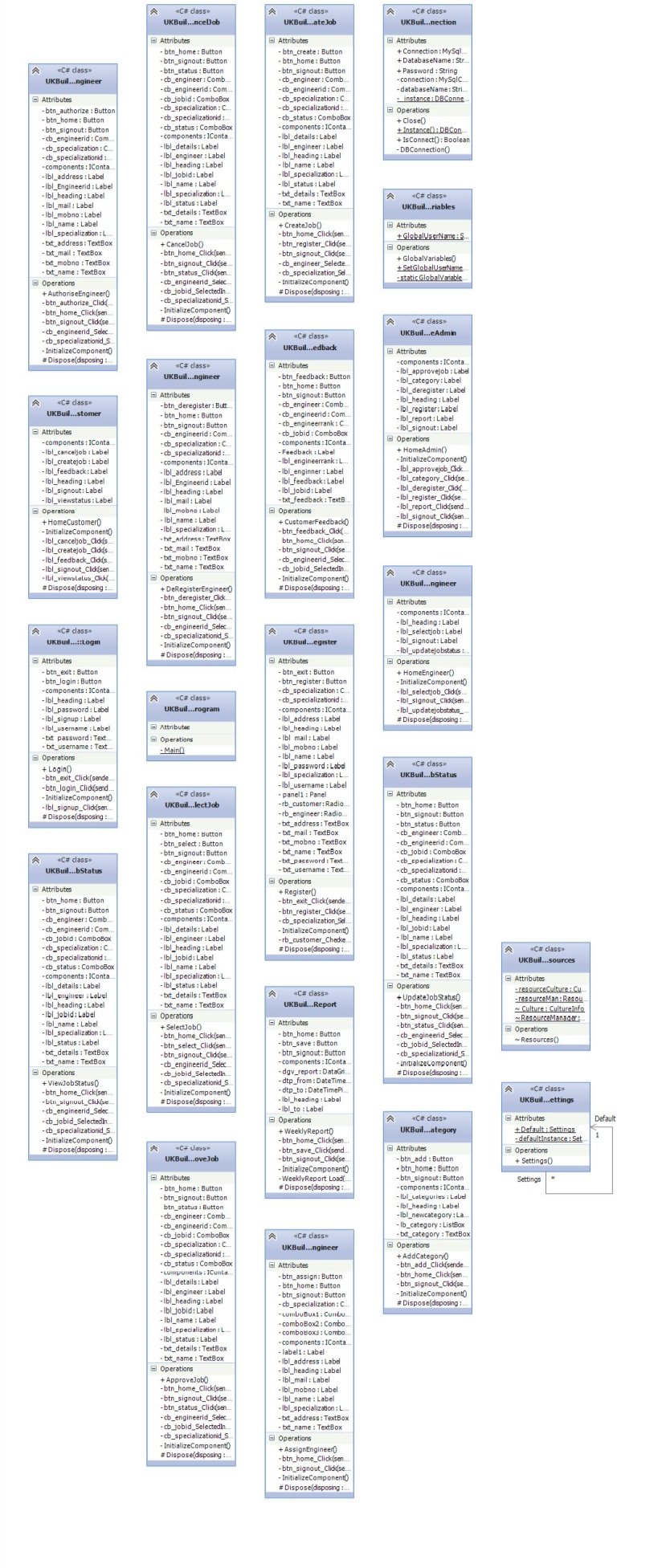


Figure 3 UML class diagram

### Class Diagram Inheritance

The below figure will show the inheritance of almost all classes to Form class, excepting Program class, DBConnection class and GlobalVariables class:



Figure 4 Class diagram for inherited classes

# System/Software Requirements-

System Requirements- it is based on the requirements to configure computer architecture. The software and hardware requirements are addressed under this heading. Memory of a computer system, operating system used, software applications, and description of processor is given. System requirement specification (SRS) is used to determine the behaviour of a system that is how a system will perform. It is important for the system designing to ensure that the user needs are satisfied. System requirements are addressed according to the functional and non-functional requirements (Inflectra, 2016).

## Functional Requirements

The Functional Requirement is the ability of a user to interact with the software application using reliable user interface design (Ofni Systems, 2017).

* Login/register- new engineers have to register themselves into the application to access the services. The existing users have to enter their valid email address and password to continue accessing with the services.
* De-register an engineer by the admin - This operation is required to eliminate the details of an engineer who no longer works with the company or was sucked. Or it can be used to eliminate any wrong entry in the system.
* Create job- There has to be an option ‘new’ to create a new entry in the system for a specified engineer. Second option should be to update the details of a particular engineer.
* Customer Feedback- customers have to login first to provide their valuable feedback for the services accepted from the company.
* Add a specialization to an engineer when he is registering
* Approve a job by the admin when it is created by a customer
* Authorise or de-authorise an engineer by the admin when the engineer registered
* Cancel an on progress or created job by a customer or by an engineer
* Generate weekly reports of jobs with all the afferent details of it
* Select job by an engineer when the job is not assigned to any engineer by the customer
* Update job status by an engineer
* View the job status by a customer

## Non-Functional Requirements

* Performance- the response time should be low so that users can access the system fatly. This affects the usability of the desktop application because more the response time will, less the customers will be interested to use the services.
* Reliability- the system has to be consistent in performance; it means that there should be less or no software failure.
* Security- authorization techniques should be used to ensure that the stored data in the system is protected from any kind of malicious activity.
* Usability- there should be no need of users to make any efforts in accessing the system. The customer service system
* Portability- system should have the capability to be accessed on different operating systems.
* Modifiability- the system should be flexible enough so that relevant changes can be made according to the demands of customers in future (Eriksson, 2012).

Description: These requirements will be taken into consideration to design a customer service system. All the listed functional and non-functional requirements will be covered while designing desktop application for UK building maintenance.

## User Requirements

**T**his determine the expectation of users for the developing a new system. It specifies the business requirements with respect to the project. User requirements are listed before the system design is created to ensure that the project will satisfy all the requirements (Ofni Systems, 2017).

Table 1 User Requirements

|  |  |  |
| --- | --- | --- |
| Role of User | Type of User | Functionality needed |
| Company Administrator | Creator | Register/deregister a new engineer  Create login page for different users like customer and engineer |
| Manager | Contributor | Create a job  Cancel a job  View progress reports |
| Engineer | Contributor | Mark completed jobs |
| Customer | Consumer | Request for service  Take service  Leave feedback |

# Software design and its specification

UML deployment diagram: it is used to represent the distribution of software as the architecture of a system. This diagram is used to visualize how the physical components are connected with each other and where the components are deployed. The purpose of these diagrams is to represent how the software components are deployed in a computer system along with the hardware components. Deployment diagram is generally made by the system engineers considering some parameters like performance, maintainability, portability and scalability of the system (Tutorials Point, 2017).

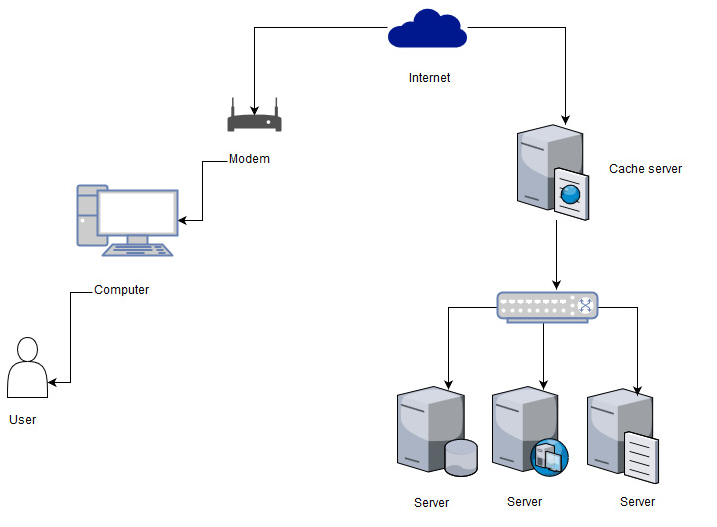


Figure 5 Deployment Diagram

The above diagram represents how the software components are deployed using hardware components. An internet cloud is connected with the router to provide internet connectivity to the computer system. User can access the software applications on the computer. On the other side of diagram, servers are attached in the network with the help of a switch. These servers will be useful to transfer the files or folders between different interconnected computers in a network.

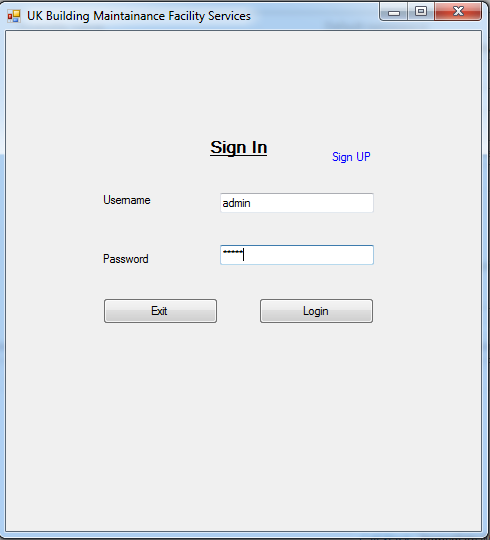


Figure 6 Login Page

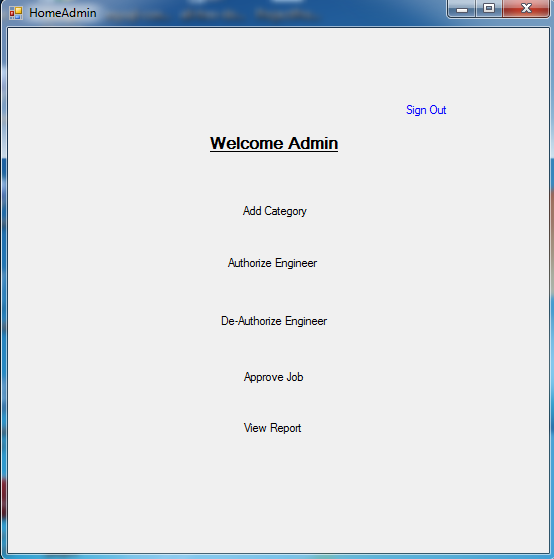


Figure 7 Successfully login

Database Design- SQL statements are used at the back-end process; these will allow users to search for a particular name or entity from the system.

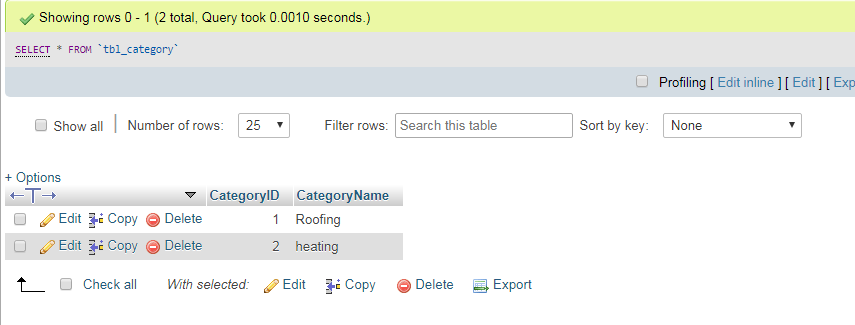


Figure 8 Database view to add category

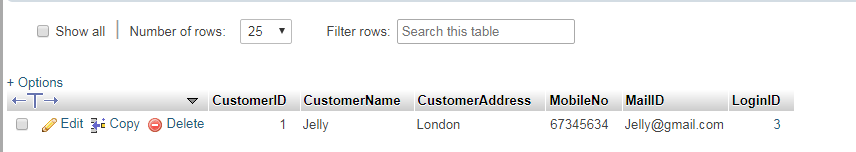


Figure 9 Customer details saved in database

# References

Eriksson, U., 2012. *Functional Requirements vs Non-Functional Requirements. [Online] ReQ test..* [Online]   
Available at: https://reqtest.com/requirements-blog/functional-vs-non-functional-requirements/  
[Accessed 10 01 2018].

Hodgson, P., 2017. *Tips for writing user manuals..* [Online]   
Available at: https://www.userfocus.co.uk/articles/usermanuals.html  
[Accessed 14 01 2018].

Inflectra, 2016. *What are System Requirements Specifications/Software (SRS)?.* [Online]   
Available at: https://www.inflectra.com/ideas/topic/requirements-definition.aspx  
[Accessed 10 01 2018].

Ofni Systems, 2017. *Functional Requirements (Functional Requirement Specifications, Functional Specs, FRS, FS).* [Online]   
Available at: http://www.ofnisystems.com/services/validation/functional-requirements/  
[Accessed 10 01 2018].

Tutorials Point, 2017. *UML - Class Diagram.* [Online]   
Available at: https://www.tutorialspoint.com/uml/uml\_class\_diagram.htm  
[Accessed 14 01 2018].

Uky, 2017. *Systems implementation..* [Online]   
Available at: http://www.uky.edu/~dsianita/695A&D/lecture5.html [Accessed 27 Dec. 2017]  
[Accessed 14 01 2018].