

TV Commercial System (Web Application)

¹Kingsley Kwanbena Ansong & ²Dr. David Ackah,

¹³Methodist University College, ²Golden Sunbeam University of Science & Technology,

Abstract

This project implements a Commercial System that will facilitate the daily business activities of Maranatha TV (GH) Ltd. It is designed to take business activities into the global world. The system allows customers to easily register with Maranatha TV (GH) Ltd and provide a mechanism for keeping track of their commercial activities using a user-friendly interface. In order to develop an efficient system to meet business requirement, a thorough research was done on the existing manual processes of the organization. This project provides the best alternative solution based on the information gathered. The TV Commercial System is a web application that makes use of latest and rich web technologies to provide an easy and effective way of managing commercial activities on TV stations. Since customers can sit at the comfort of their offices and do business with the company, the TV commercial system will go a long way to reduce drastically the cost of certain business operations such as reduce office space requirements, maintain sizeable employees, reduce travelling expenses, etc.

Keyword: TV Commercial, Web Application, Commercial Media, Media Marketing

I. INTRODUCTION

According to (Taflinger, 1996), Advertising is the non-personal communication of information usually paid for and usually persuasive in nature about products, services or ideas by identified sponsors through the various media. Advertising on television usually means running a TV commercial, typically between 10 and 60 seconds in length, but TV advertising can also include sponsorships, announcement, and product placement within television shows. This chapter describes the overview of the TV Commercial system. It outlines the scope, aims and objectives of the project. It also describes the problem definition and the justification for undertaking this project. This report also contains discussions of the fact finding techniques used in eliciting information as well as the development methodology adopted in undertaking this project. This document presents progress made so far in the attempt of computerizing some of the activities carried out in the operations of the Maranatha TV (GH) Ltd. Maranatha TV Limited is a private television broadcaster. The company began on-air transmission on December 12th 2008 and produces a variety of television programs including news, drama, reality and entertainment shows. Maranatha TV offers to its cherished viewers the most exciting programs in terms of creating, developing, producing, and broadcasting innovative and quality television content. This has propelled Maranatha TV to become a household name in Ghana and beyond. The overwhelming increase in customer base has necessitated the need for a computerized system to handle some of the main activities of the company effectively since the current manual system poses many challenges.

II. SPECIFICATION

A. Introduction

This outlines a detailed description of the requirements of the TV Commercial System. It covers how the entire requirements model of the system was conducted. It Explains the features of the system, the interfaces of the system, what the system will do and the constraints under which it must operate. The functional requirements are documented using use cases.

B. Analysis

This section mainly deals with activities used to capture and model the requirements which will serve as the basis for additional development. Specification, being the final product produced by the requirements engineer, according to (Pressman, 2005), serves as the foundation for subsequent software engineering activities. It explains the function and performance of computer based system and the constraint that will govern its development.

i. Intended Users

This document is designed for use by the project stakeholders including end users of the system (Manager, Employee, and Customer), testers and the development team. This document will be reviewed regularly whenever there is a new release (increment). In the case of any changes in the user requirements, this document will be updated to reflect the changes accordingly.

ii. Product Scope

The TV Commercial System is a web application which is developed to carry-out and manage the core activities of Maranatha TV (GH) Ltd. The TV Commercial System will eliminate the manual way of managing customers by providing a user-friendly interface to add new customer records and also search and retrieve customer details with ease. The system is designed to maximize productivity by providing different ways of monitoring or tracking various activities such as booking, scheduling, and commercials played. By maximizing users' work and efficiency, the system will meet the needs of the user and therefore remains beneficial.

C. General System Description

i. System Environment (Product perspective)

This section of the document shows the boundary or scope of the system and also the relationship between the system and its environment. Below is the diagram.

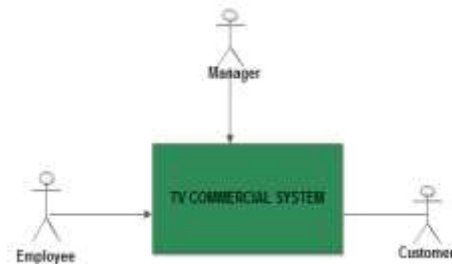


Figure 2.1 – Overall system description

ii. User Classes and Characteristics

User Class	Description
Administrator/Manager	This user class will have full system privilege and permissions. He can add and remove users, add and update commercial schedule, grant and revoke permissions of users. He performs all the functions performed by the other users of the system
Employee	Employee user class will use the system to perform various functions such as viewing customer commercial activities, view commercial schedule, add/update commercials played, etc.
Customer	The customer will be given a full access to all customer functions. Some of the functions that the customer can perform include adding new product, contract details, booking for commercials etc.
Other Stakeholders	Mr. Gary Budgen, Project Supervisor

D. Requirement Specification

This section gives detailed description of the functional requirements, non-functional requirements, and the interface requirements of the TV Commercial System. This is critical to the overall success of the project development process. According to (Sommerville, Software Engineering, 2007), the requirements of a system are the description of the services provided by the system and its operational constraints. These requirements reflect the needs of customers for a system that helps solve some problems.

E. Interface Requirements

i. User Interface

The system will make use of buttons and graphical icons to deliver user-friendly solution to its users. Examples of objects that will be used in the program development include;

- Buttons
- List boxes
- Drop down menus
- Text boxes, and
- Labels

The system will have different interfaces including the following

- Registration interface for customers to provide their registration information
- Login interface for all users with the correct username and password to login.
- Different interfaces to facilitate the management of commercials.
- Other interfaces here will include contract, bookings, rate, product, programs, and commercial type interfaces.

See appendix B for the screenshots of the user interfaces of the system

ii. Hardware Interfaces

Server side configurations

- Operating System: - Windows OS
- Memory: - A minimum of 4 GB
- Hard drive: - A minimum of 20GB free disk space

Client side configurations

- Operating System: - Windows/Linux. Preferably Windows XP/7
- Memory: - A minimum of 512 MB
- Hard drive: - A minimum of 10GB free space

iii. Software Interfaces

The product was developed using Dreamweaver, PHP and Mysql. The end product will be a web based application

iv. Communication Interfaces

Users must have a web browser such as Mozilla Firefox, Netscape, Internet explorer, Opera as the communication interface.

F. Functional Requirements

The functional requirements according to (Sommerville, Software Engineering, 2007), are the 'services a system should provide, how the system should react to particular inputs and how the system should behave in particular situations'. The TV Commercial System will enable users of the system to perform functions (such as write, modify, read and delete) records. The product is designed to enable users to carry out their assigned tasks efficiently and accurately. To show the various functions of the TV Commercial System, use case diagrams will be used to show how these various functions are undertaken by the system. Actors in the form of stick figures will be used to show various roles that a user can play. A user can have one or more roles. The use case for the TV Commercial System is shown below.

i. High-Level Use Case Analysis

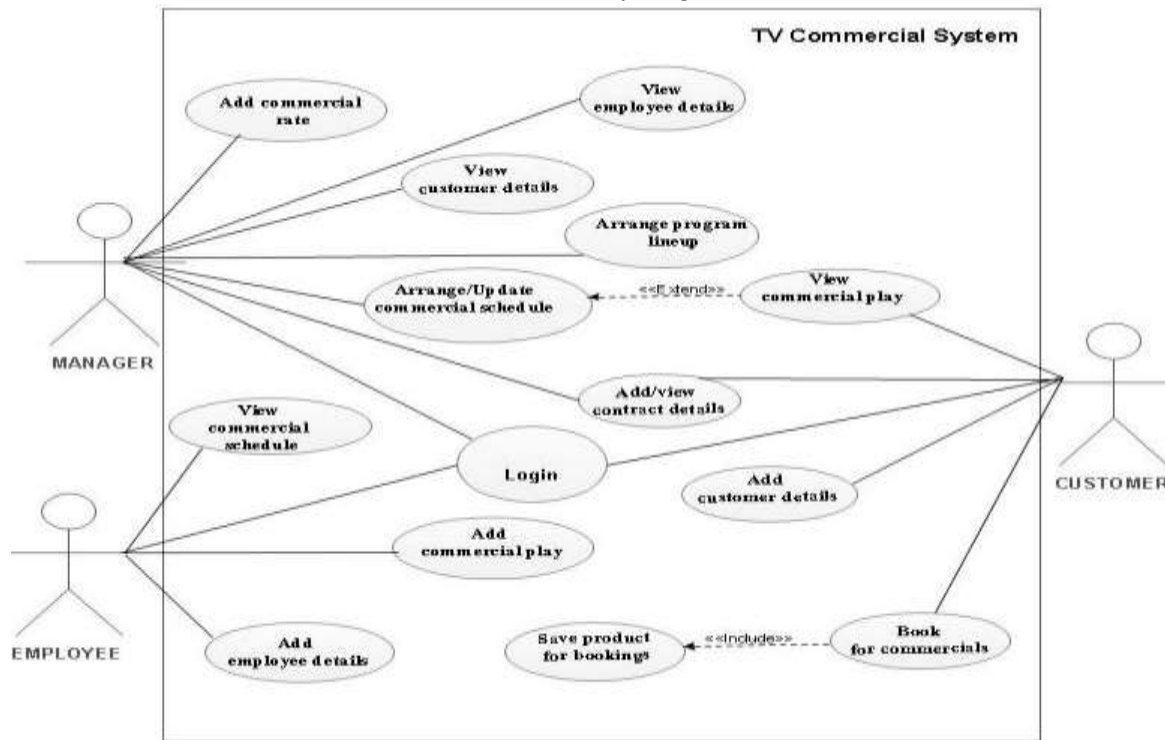


Fig. 2.2 - TV Commercial System (High-Level Use Case Diagram)

ii. *Detailed Use Case Description (Log on to System)*

Description and Priority: This is used to enter the system and gain access to system functionalities. System functionalities are accessible based on user permissions. All users of the system would first have to log on to the system and be granted access before they can perform any functions. This is of higher priority because it deals with a security issue.

Stimulus/Response Sequences: Any user who wishes to use the system to perform any function would have to provide the correct credentials to log on to the system.

Functional Requirements: REQ 1 - If the user leaves any of the required fields blank; the system prompts the user to enter his credentials. REQ 2 - The system will deny access and request for a retry if a user enters one correct credential but the remaining fields are entered wrongly, for example correct username but wrong password. REQ 3 - If a user provides wrong credentials for all fields, the user is denied access. REQ 4 - A user will only be granted access if all his credentials are correct in all fields.

Detailed Use Case Description



Fig. 2.3 – Use case diagram for Login to system

Use Case Name	Login
Actor(s)	Manager, Customer, and Employee

Trigger	User opens the application
Precondition	None
Basic Path	<ol style="list-style-type: none"> 1. User clicks on 'login' 2. The system displays login form and then prompt user to enter their details (username and password) 3. User enters the right credentials (username and password). 4. System authenticates and allow user access to permitted functionalities
Alternative Paths	<ol style="list-style-type: none"> 1. User enters wrong details (username and /or password). 2. Message box pops up to indicate that the user credentials entered were not correct and prompt the user to enter the correct details (username and password).
Post-condition	Able to use other system functionalities
Exception Paths	User may choose to logout at any time

Add a new customer

- **Description and Priority:** This is used to capture and store the details of a new customer. The Customer as well as the Manager has the right to add, view, update or delete all user details. This is of high priority because it serves as one of the main reasons of undertaking this project.
- **Stimulus/Response Sequences:** The user will click on the register button to open the registration form. Upon completing the fields correctly, he/she will click on the save button to add user to the database. The system will display a message box to that effect.
- **Functional Requirements:** REQ 1 - The values for the various fields must be provided and also ensure that they are correct. If any field is left blank, the system will prompt users to fill that field before saving. REQ 2 - If correct details are entered, the save button is clicked and the user details is saved in the database.

Detailed Use Case Description



Fig. 2.4 – Use case diagram for add a New Customer

Use Case Name	Add customer
Actor(s)	Customer, Manager
Trigger	New customer want to book for commercials
Precondition	None
Basic Path	<ol style="list-style-type: none"> 1. User starts (opens) the application 2. User selects the option to add customer 3. System displays form to add a new customer 4. User enters required information and saves 5. System validates and confirms information.
Alternative Paths	None
Post-condition	New customer details captured and saved
Exception Paths	User may choose to logout at any time

iii. Add Contract

- **Description and Priority**

This module allows the users (manager, and customer) to add contract details. It will contain information such as the customer details including product, start and end dates of contract. It will also allow for contract modifications. This is of higher priority since without contract there can be no bookings for commercials.

- **Stimulus/Response Sequences**

After the customer has provided the correct log in credentials and given access to the main menu interface, the customer enters contract details.

- **Functional Requirements**

REQ 1 – the customer must login to the system before entering contract details. If the customer has not logged in to the system, the system prompts for login before proceeding.

REQ 2 - If the user leaves any of the required fields blank; the system prompts the user to fill all the fields.

REQ 3 - A user will only be granted access if all his credentials are correct.

Detailed Use Case Description

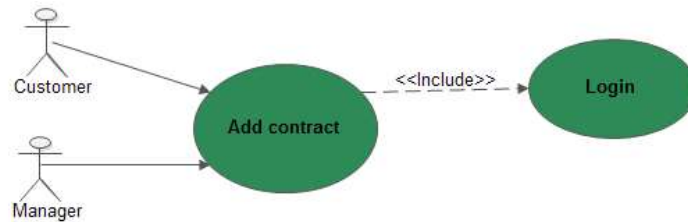


Fig. 2.5 – Use case diagram for Add contract

Use Case Name	Add contract details
Actors	Customer and Manager
Trigger	Customer clicks on the ‘General customer activities’ button
Precondition	The customer must be Logged on to the system before proceeding to perform this operation
Basic Path	<ol style="list-style-type: none"> 1. User start (open) the application 2. User selects the option to add product 3. System displays form to add new product details 4. User enters required information and saves 5. System returns confirmation information
Alternative Paths	<ol style="list-style-type: none"> 1. Customer views ‘contract details 2. Customer modifies contract details 3. Customer cancels/deletes contract details
Post-condition	Contract details are captured and stored.
Exception Paths	User may choose to logout at any time

iv. Add Product

- **Description and Priority**

This module is used to capture and store specific product details. The customer has the right to add, update or delete product details. Managing customer product is highly prioritized because it is part of the reasons for designing this system.

- **Stimulus/Response Sequences**

After the customer has provided the correct log in credentials and given access to the main menu interface, he/she will click the ‘General customer activities’ button and then click again on the ‘product’ button to open the product form. The customer enters the right product details and clicks the save button to save product. The system will then display a message box showing it was successful.

- **Functional Requirements**

REQ 1 – the customer must login to the system before entering contract details. If the customer has not logged in to the system, the system prompts for login before proceeding.

REQ 2 - If the user leaves any of the required fields blank; the system prompts the user to fill all the fields.

REQ 3 - A user will only be granted access if all his credentials are correct.

Detailed Use Case Description



Fig. 2.6 – Use case diagram for Add product
Add product

Use Case Name	Add product
Actor(s)	Customer
Trigger	Customer clicks on the ‘General customer activities’ button
Precondition	The customer must have already logged on to the system and also be in contract.
Basic Path	<ol style="list-style-type: none"> 1. User clicks the ‘General Customer Activity’ button. 2. The System displays the list of activity buttons. 3. Customer clicks on the ‘Product’ button. 4. The Product form is displayed with buttons adding new Product, exit, remove etc. 5. customer click on the add new button 6. User enters the details and clicks save. 7. System displays message box to confirm the save action has been successful.
Alternative Paths	<ol style="list-style-type: none"> 1. Customer views product details. 2. Customer updates product details.
Post-condition	Product details captured and saved
Exception Paths	User may choose to logout at any time

v. Add/Update Bookings

- **Description and Priority**

This module is used to capture and store details of number of bookings by customers. The customer as well as the Manager has the right to add, modify or cancel bookings. Managing customer bookings is highly prioritized because it is part of the reasons for designing this system.

- **Stimulus/Response Sequences**

After the customer or the Manager has provided the correct login credentials and given access to the main menu interface, he/she will click the ‘General customer activities’ button and then click again on the ‘Booking’ button to open the form for booking. The customer enters and selects choices before clicking the save button to save booking. The system will then display a message box showing it was successful.

• **Functional Requirements**

- REQ 1 – the customer must login to the system before entering contract details. If the customer has not logged in to the system, the system prompts for login before proceeding.
 REQ 2 – the customer must have an existing contract: If the customer’s contract has expired, the system denies the customer from booking for commercials
 REQ 3 - If the user leaves any of the required fields blank; the system prompts the user to fill all the fields.
 REQ 4 - A user will only be granted access if all his credentials are correct.

Detailed Use Case Description

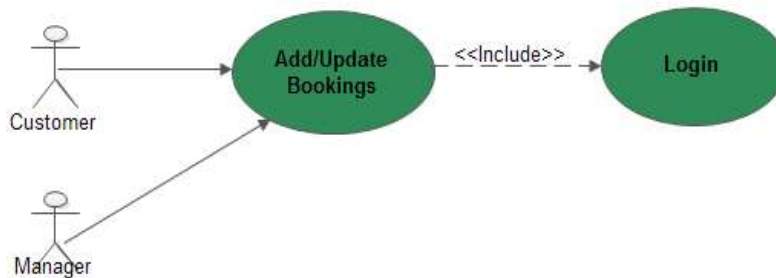


Fig. 2.7– Use case diagram for Add product

Use Case Name	Add/update bookings
Actor(s)	Customer, Manager
Trigger	User clicks the ‘General Customer Activity’
Precondition	The customer must have already logged on to the system and also be in contract.
Basic Path	<ol style="list-style-type: none"> 1. User clicks the ‘General Customer Activity’ button. 2. The System displays the list of activity buttons. 3. Customer clicks on the ‘Booking’ button. 4. The system displays ‘booking’ form with buttons for adding new, modify, remove booking etc. 5. customer click on the add new button 6. User enters/ selects the details and clicks save. 7. System displays message box to confirm the save action has been successful.
Alternative Paths	<ol style="list-style-type: none"> 1. User views booking details. 2. Customer modifies booking details. 3. customer cancel bookings
Post-condition	Commercial bookings captured and saved
Exception Paths	User may choose to logout at any time

vi. **Add commercial schedule**

• **Description and Priority**

This module is used to capture and store commercial schedule details. The Manager has the right to add, modify or delete commercial schedule. Managing customer bookings is highly prioritized because it is part of the reasons for designing this system.

• **Stimulus/Response Sequences**

After the Manager has provided the correct login credentials and given access to the main menu interface, he/she clicks the ‘ General Internal Operations’ button, then open the commercial activities form by clicking the ‘Commercial’ button. The manager clicks on the ‘Schedule’ button to open the commercial schedule form. The manager enters and selects schedule details and clicks the save button to save. The system will then display a message box showing it was successful.

• **Functional Requirements**

REQ 1 – the manager must be logged on to the system before entering commercial schedule details. If the manager has not logged in to the system, the system prompts for login before proceeding.

REQ 2 - If the manager leaves any of the required fields blank; the system prompts the manager to fill all the fields.

REQ 3 - A user will only be granted access if all his credentials are correct.

Detailed Use Case Description

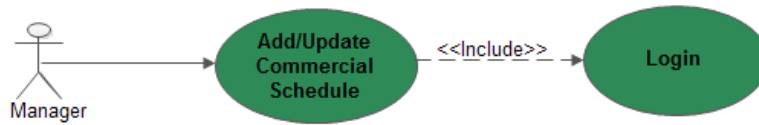


Fig. 2.8 – Use case diagram for Add product

Add/Update commercial schedule

Use Case Name	Add/update commercial schedule
Actor(s)	Manager
Trigger	Manager clicks on ‘General Internal Operation’
Precondition	The customer must have already logged on to the system and also be in contract.
Basic Path	<ol style="list-style-type: none"> 1. Manager clicks the ‘General Internal Operations’ button. 2. The System displays the list of internal operation buttons. 3. Manager clicks on the ‘Commercial Schedule’ button. 4. The system displays ‘Commercial Schedule’ form with buttons for adding new, modify, remove etc. 5. Manager clicks on the add new button 6. Manager enters schedule details and clicks save. 7. System displays message box to confirm the save action has been successful.
Alternative Paths	<ol style="list-style-type: none"> 1. Manager views commercial schedule details. 2. Manager modifies commercial schedule. 3. Manager cancels commercial schedule details.
Post-condition	Commercial Schedule captured and saved
Exception Paths	User may choose to logout at any time

vii. Add Program Schedule

• Description and Priority

This module is used to capture and store General program schedule details. The Manager has the right to add, modify or delete program schedule. This is of higher priority because it is part of the reasons for designing this system.

• Stimulus/Response Sequences

After the Manager has provided the correct login credentials and given access to the main menu interface, he/she clicks the ‘ General Internal Operations’ button, then open the program form by clicking the ‘Program’ button.. The manager enters the Program details and clicks the save button to save. The system will then display a message box showing it was successful.

• Functional Requirements

REQ 1 – the manager must be logged on to the system before entering commercial schedule details. If the manager has not logged in to the system, the system prompts for login before proceeding.

REQ 2 - If the manager leaves any of the required fields blank; the system prompts the manager to fill all the fields.

Detailed Use Case Description

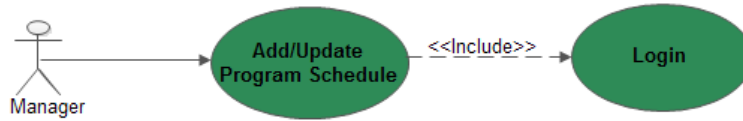


Fig. 2.9 – Use case diagram for Add product

Add/Update Program Schedule

Use Case Name	Add/update program schedule
Actor(s)	Manager
Trigger	Manager clicks on ‘General Internal Operation’
Precondition	The Manager must have already logged on to the system.
Basic Path	<ol style="list-style-type: none"> 1. Manager clicks the ‘General Internal Operations’ button. 2. The System displays the list of internal operation buttons. 3. Manager clicks on the ‘Program Schedule’ button. 4. The system displays ‘Program Schedule’ form with buttons for add new, modify, remove etc. 5. Manager clicks on the add new button 6. Manager enters program details and clicks save. 7. System displays message box to confirm the save action has been successful.
Alternative Paths	<ol style="list-style-type: none"> 1. Manager views program schedule details. 2. Manager modifies program schedule. 3. Manager cancels program schedule details.
Post-condition	Program Schedule captured and saved
Exception Paths	User may choose to logout at any time

viii. Add commercial play

• **Description and Priority**

This module is used to capture and store Commercial played. The Employee as well as the Manager has the right to add, modify or delete commercial played. This is of higher priority because it helps customers to track their commercials.

• **Stimulus/Response Sequences**

After the Employee or the Manager has provided the correct login credentials and given access to the main menu interface, he/she clicks the ‘ General Internal Operations’ button, then open the Commercial play form by clicking the ‘Commercial’ button. The Employee enters the Commercial details that have been played and clicks the save button to save. The system will then display a message box showing it was successful.

• **Functional Requirements**

REQ 1 – the Employee must be logged on to the system before entering Commercial played details. If the Employee has not logged in to the system, the system prompts for login before proceeding.

REQ 2 - If the Employee leaves any of the required fields blank; the system prompts him/her to fill all the fields.

Detailed Use Case Description

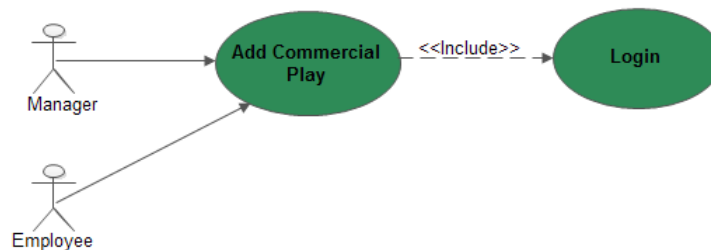


Fig. 2.10 – Use case diagram for Add product

Add Commercial play

Use Case Name	Add Commercial play
Actor(s)	Employee, Manager
Trigger	Employee clicks on ‘General Internal Operation’
Precondition	The Employee must have already logged on to the system.
Basic Path	<ol style="list-style-type: none"> 1. Employee clicks the ‘General Internal Operations’ button. 2. The System displays the list of internal operation buttons. 3. Employee clicks on the ‘Commercial play’ button. 4. The system displays ‘Commercial play’ form with buttons for add new or remove. 5. Employee clicks on the add new button 6. Employee enters Commercial play details and clicks save. 7. System displays message box to confirm the save action has been successful.
Alternative Paths	<ol style="list-style-type: none"> 1. Employee views Commercial play details. 2. Employee deletes Commercial play details.
Post-condition	Commercial play details captured and saved
Exception Paths	User may choose to logout at any time

vix. Commercial Rate

• **Description and Priority**

This module is used to capture and store Commercial rate details. The Manager has the right to add, modify or delete commercial rate. This is of medium priority because it helps customers to choose the best commercial type base on their financial strength.

• **Stimulus/Response Sequences**

After the Manager has provided the correct login credentials and given access to the main menu interface, he/she clicks the ‘ General Internal Operations’ button, then opens the Commercial rate form by clicking the ‘Commercial rate’ button. The Manager enters the Commercial rate details and clicks the save button to save. The system will then display a message box showing it was successful.

• **Functional Requirements**

REQ 1 – the Manager must be logged on to the system before entering Commercial played details. If the Employee has not logged in to the system, the system prompts for login before proceeding.

REQ 2 - If the manager leaves any of the required fields blank; the system prompts him/her to fill all the fields.

Detailed Use Case Description

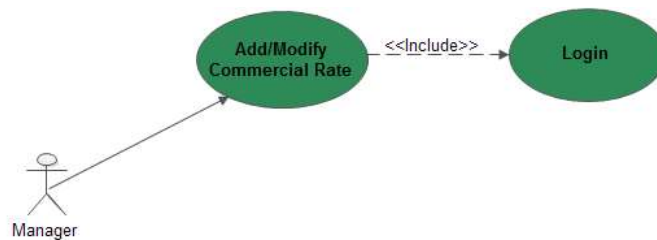


Fig. 2.11 – Use case diagram for Add product

Add Commercial Rate

Use Case Name	Add Commercial rate
Actor(s)	Manager
Trigger	Manager clicks on ‘General Internal Operation’
Precondition	The Employee must have already logged on to the system.

Basic Path	<ol style="list-style-type: none"> 1. Employee clicks the ‘General Internal Operations’ button. 2. The System displays the list of internal operation buttons. 3. Manager clicks on the ‘Commercial rate’ button. 4. The system displays ‘Commercial rate’ form with buttons for add new, modify, and delete. 5. Manager clicks on the add new button 6. Manager enters Commercial play details and clicks save. 7. System displays message box to confirm the save action has been successful.
Alternative Paths	<ol style="list-style-type: none"> 1. Manager views Commercial rate details. 2. Manager modifies Commercial rate details. 3. Manager deletes commercial rate details
Post-condition	Commercial rate details captured and saved
Exception Paths	User may choose to logout at any time

G. Non-Functional Requirements

The non-functional requirements are those that are not directly concerned with the specific functions delivered by the system as stated by (Sommerville, Software Engineering, 2007). These do not usually just apply to individual system features or functions but apply to the system as a whole. This will include system properties such as performance, usability, security, reliability and ease of maintenance. Others may include some of the processes that should be used to develop the system, tools to be used and some standards that should be met during the development process.

i. Software Quality Attributes

Quality attributes are the overall factors that affect run-time behavior, system design, and user experience. They represent areas of concern that have the potential for application wide impact across layers and tiers. Some of these attributes are related to the overall system design, while others are specific to run time, design time, or user centric issues. The extent to which the application possesses a desired combination of quality attributes such as usability, performance, reliability, and security indicates the success of the design and the overall quality of the software application.

ii. Performance Requirements

This provides an indication of the responsiveness of the system in execution. The system must be in the position to handle the anticipated amount of data including system through put and storage. In enhancing the TV Commercial System performance, messages are in place to serve as a guide to indicate to users for successful or unsuccessful operations.

iii. Usability Requirements

This describes how well the system meets the requirements of the user by being intuitive, and easy to customize to their preference. Providing this functions in the system results in overall user experience and ensures that users are comfortable when using the system.

- The System will provide flexible and easy to use interface that can be learned in a short Period of time by the user.
- The System shall keep fast response times to user interaction
- The system will offer the ability to operate effectively within any major web browser, such as Google chrome, Internet Explorer and Mozilla Firefox.
- The system will allow a user to cancel an action and navigate forward or backward within the application links.

iv. Maintenance Requirements

Maintainability deals with the ability of the system to successfully undergo changes with flexibility and ease. These changes could impact services, features, and interfaces of the system. The codes are clear and concise, thus making the software more understandable and flexible. The system is developed as a structure of highly independent modules and allows future updates and modifications.

v. Security Requirements

This system was designed with security features in mind. The application will implement security features in the form of username and password. Data will be protected from unauthorized users by the implementation of an authentication mechanism. Adhering to security considerations ensures that every user is authenticated before he/she will be allowed to proceed to perform any functions. For example, customer must be logged on to the system before he/she can enter booking details. Another area of security concern has to do with securing the database. In order to stay secured, a copy of the database will be periodically taken onto an offline storage media. This will be used to assist in data recovery in case of a database failure.

vi. Constraints

Some constraints were placed on the application. These are listed below

1	User can not have the same user name in the system
2	A customer must be in contract with Maranatha TV before he/she can book for commercials
3	User needs to login and be authenticated before access is granted to any further functions in the system

H. Assumptions

It is assumed that all the necessary requirements needed for the smooth and effective running of this system has been put in place. These requirements include the minimum requirement of computer resources needed to run this program.

i. References

A portion of the interim report and initial proposal was reused in the specification. This includes diagrams, functional and non-functional requirements of the proposed system

ii. Summary

This chapter produced an overview of the requirement specification for the system. Interface, functional and non-functional requirements were specified. UML use case diagrams and use case descriptions were used to describe some of the functionalities of the system. It gave a brief overview on the objects used in designing the user interface. It ended with a brief discussion on some non-functional requirement and software quality attributes like performance requirement, usability requirement, maintenance requirements and security requirements.

III. DESIGN

This chapter describes the overview of the overall system design. The chapter explains the various levels of design and applies them to the project. These are system architectural design, data design and system interface design. The way each use case is realized (use case realization) is also covered here. The architectural design in this chapter will describe functions relating to the architecture of the overall system. The system will be looked at from different perspectives such as Interaction view, use case view, and deployment view. Various UML diagrams would be used to model these views. Data design covering Enhanced Entity Relation Diagram as well as data objects and data dictionary will also be discussed here. Finally, the discussion of interface design with screenshots of some of the user interface of the system.

A. Architectural Design

The architectural design describes the sub-systems that make up the TV Commercial system and the relationship between them. System architectural design according to (Sommerville, Software Engineering, 2004) describes the design process for identifying the sub-systems making up a system and the framework for sub-system control and communication. The design stages helps in specifying the hardware and software requirements and also define the overall system architecture. This section of the document will look at the system from different perspectives. The system will be modeled from the viewpoints discussed below:

- Use case view
- Interaction view – Use case realization
- Deployment view

i. Use Case View

Use-case diagrams are drawn up during the preliminary analysis phase of a project to record the interactions of the proposed system with users and other external systems. In effect, use case diagrams provide a high-level view of what the system does and how it is used. The users, their functional requirements and the associations between all the major parts, can be clearly exposed. The use case diagram does not carry detailed information about the sequencing of

actions; this behavioral information can only be noted in associated text which can be very useful when subsequently drawing up sequence diagrams.

The concern is at a higher level of abstraction with no immediate view to implementation. However, the identification of actors, in the use case diagram, may assist later on with the specification and implementation of system interfaces. (Williams, 2006). A set of use cases which describe all possible interactions in the TV Commercial System has already been taken care of in **Fig. 2.2** (use case diagram) of the Specification.

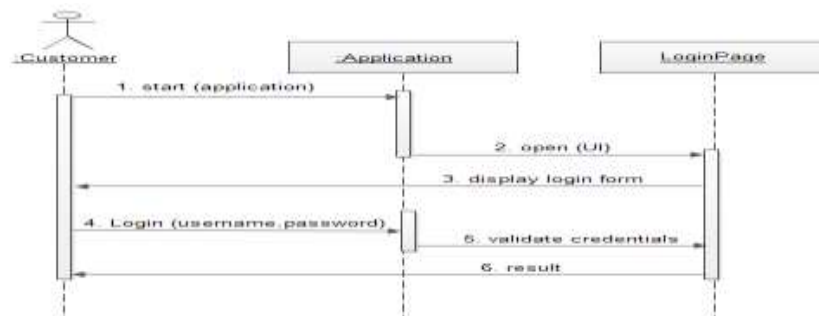
ii. Interaction View – Sequence Diagrams

Sequence diagram shows the interaction between users, systems and sub-systems, and also emphasize the ordering of time of messages.

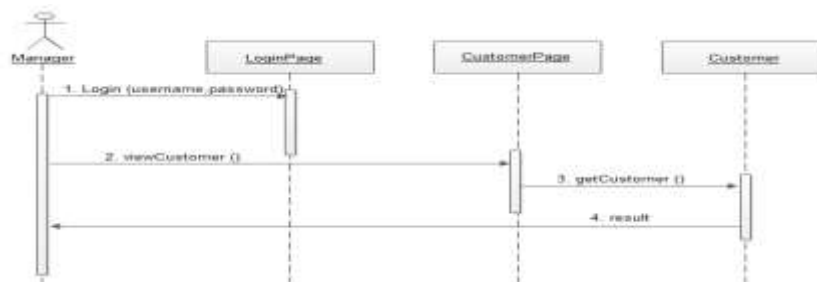
Below is the section that shows interactions between the various objects in the TV Commercial System.

Use Case Realizations

- **Login to system**



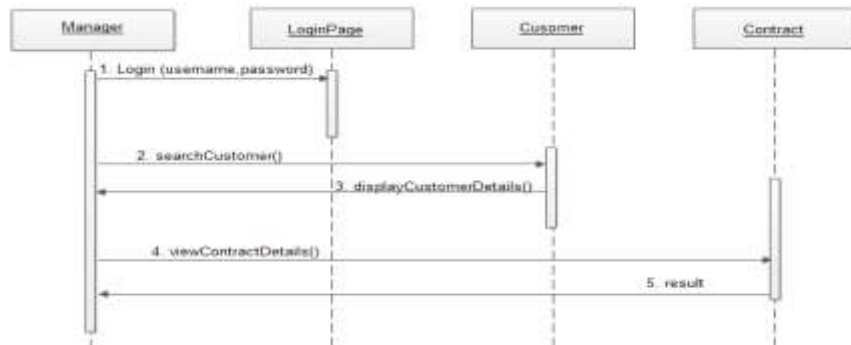
- **View customer details**



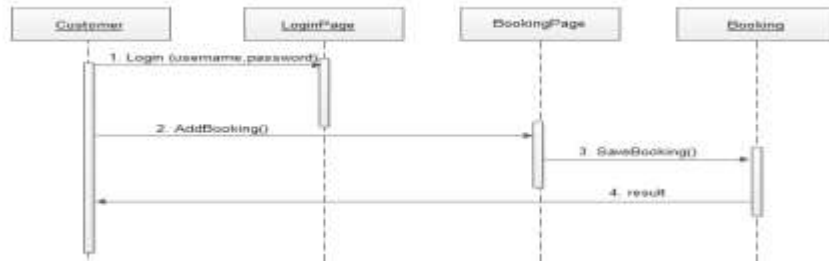
- **Add Product**



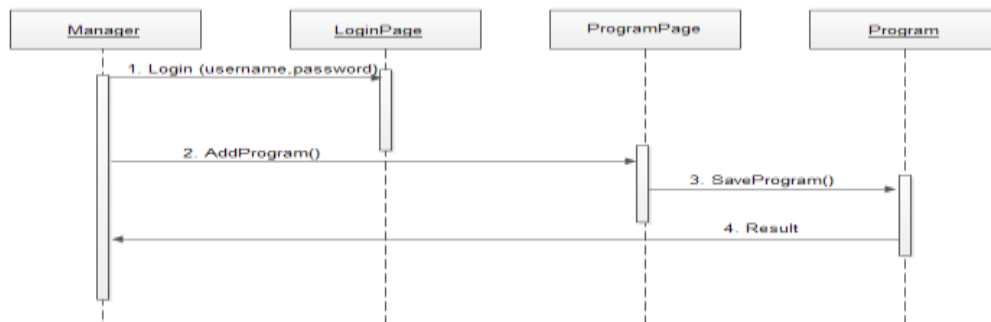
- **View customer contract**



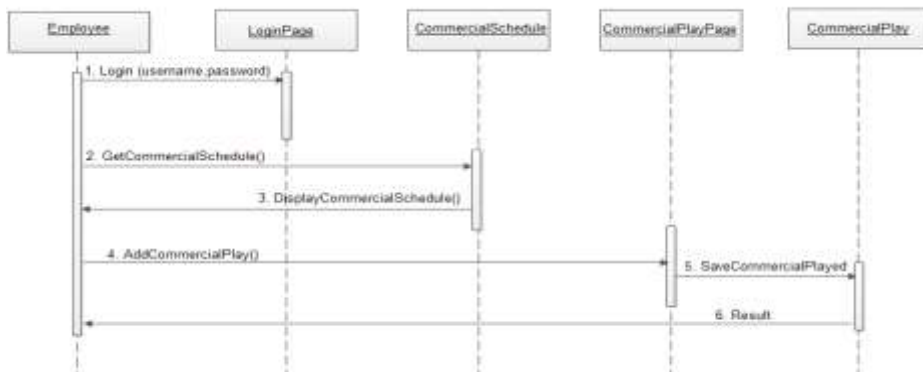
- **Add commercial booking**



- **Add Program Schedule**



- **Add Commercial Played**



iii. **Deployment View – Deployment Diagram**

Deployment diagram used in developing the TV Commercial system shows the relationships between the software and hardware components in the system and the physical distribution of the processing. According to (IBM, 2004), deployment diagram during the implementation phase of development shows the physical arrangement of the nodes

in a distributed system, the artifacts that are stored on each node and then the components and other elements that the artifacts implement. Communication paths and relationships model the connections in the system. Deployment diagrams are effective for visualizing, specifying, and documenting different types of systems such as the TV Commercial System. Because deployment diagrams focus on the configuration of the runtime processing nodes and their components and artifacts, it is possible to use this type of diagram to assess the implications of distribution and resource allocations.

Below is the deployment diagram of the system.

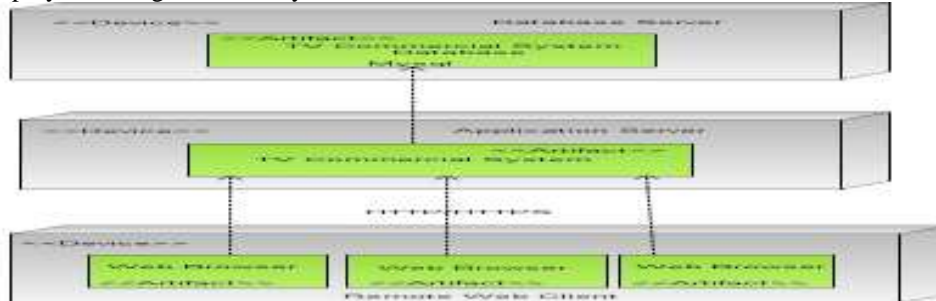


Fig – 3.1 - Deployment Diagram

B. Data Structure Design

Data structure as described by (Rouse, 2006) is a specialized format for organizing and storing data. It is an interrelationship among data elements that determines how data is kept, manipulated and presented.

In designing data structure for this project, this section covers Enhanced Entity Relation Diagram as well as data objects and data dictionary of the TV Commercial System.

i. Enhanced Entity Relationship Diagram

Enhanced ERD is an extended or enhanced version of ERD. It contains the Semantic concepts incorporated into the original ER model. Providing more information about the entities in the TV Commercial System, the Enhanced ERD utilizes the specialization/generalization concept. According to (Connolly & Begg, 2005) Specialization is the process of maximizing differences between members of an entity by identifying their distinguishing characteristics. This is achieved via sub class which handles distinct sub grouping of occurrences of an entity type.

Generalization is the process of minimizing differences between entities by identifying their common characteristics. This is achieved via Super class in EER modeling. Super class is an entity type that includes one or more distinct sub groupings of its occurrences. In developing the TV Commercial system, the two main rules in the EERD were considered and taken into account. They are:

- Participation constraint which determines whether every member in the super class must participate as a member of a subclass.
- Disjoint constraint describes the relationship between members of the subclasses and indicates whether it is possible for a member of a super class to be a member of one, or more than one, subclass.

Below is the Enhance Entity Relationship of the TV Commercial System

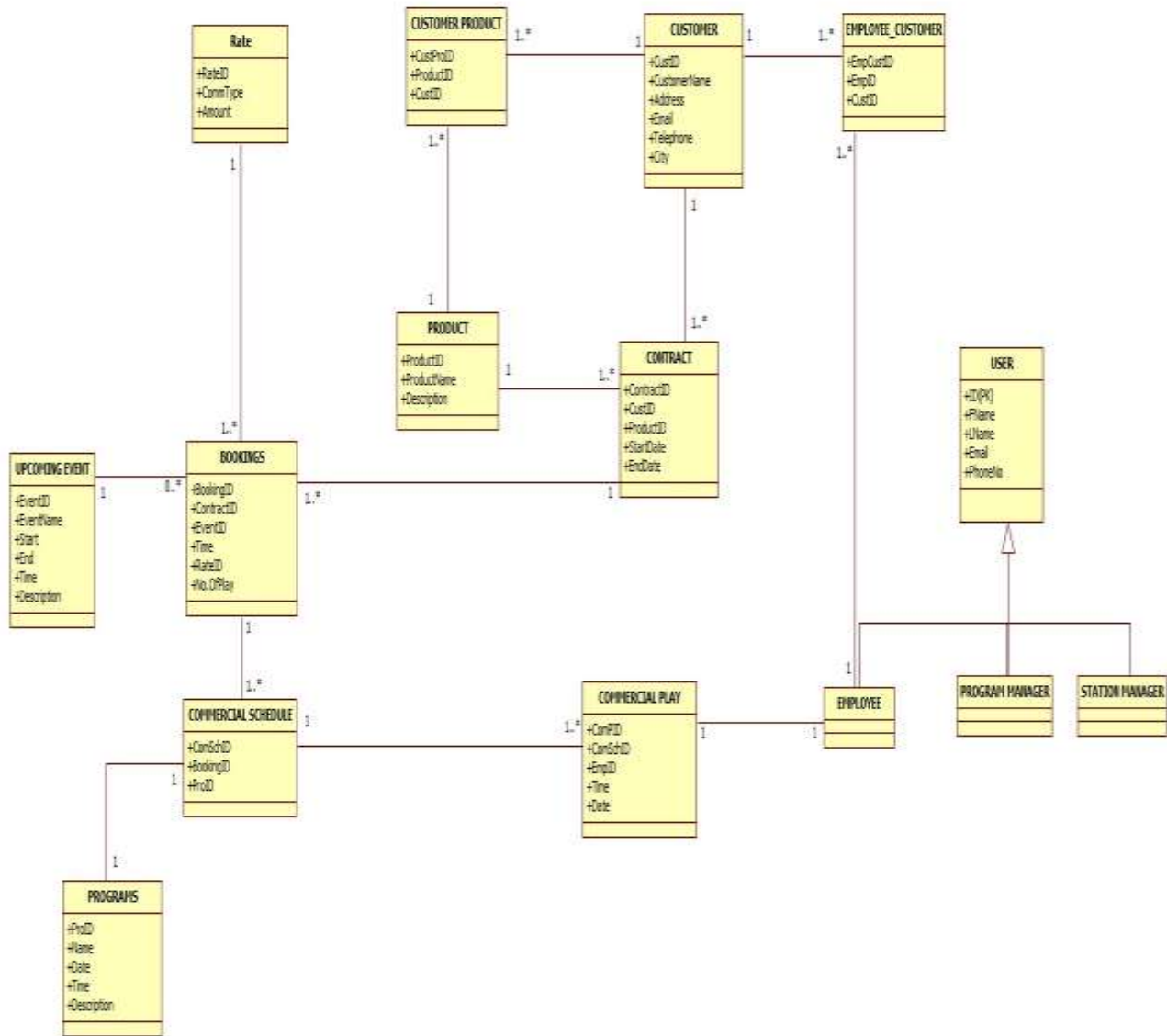


Fig 3.2 – Enhance Entity Relationship Diagram

ii. Data Objects and Data Dictionary

Data dictionaries help design databases consistently and satisfy system documentation requirements. As outlined by (Francl, 2007) Creating the dictionary during the early systems design phase enhances design activities by identifying individual data elements and where they are stored in the system. This organized approach saves time and eliminates boring programming tasks. Afterward, system maintenance is easier because the dictionary forces discipline of a single definition for each data entity. A data dictionary contains a list of all files in the database and the names and types of each field. **Table 3.3.2** below shows a brief description of the main data objects or entities data that make up the TV Commercial System.

Data object/Entity	Description
User	This entity holds information about the users of the system.

Customer	This entity holds personal/corporate information about all the customers of the Maranatha TV Ghana Ltd.
Product	This entity holds information about product details that have been submitted by customers for commercials.
Contract	This entity holds information about contract details between customers and Maranatha TV Gh. Ltd. The data include when the contract starts and when it ends.
Booking	This entity holds information about bookings made so far by customers.
Program Schedule	This entity holds information about programs of the TV station that are scheduled so that customer can view and book for their commercials around that time.
Upcoming Event	This entity holds information about upcoming programs that are not yet scheduled. The data include occasional programs that are coming up at later date such as upcoming football tournament, Olympics games, Xmas shows, etc
Commercial Schedule	This entity holds information about customer product that has been put on schedule to be played at the specified time.
Rate	This entity holds information about each type of commercials and their rate.
Commercial play	This entity holds information about customer product that have been played

A. System Interface Design

It is an open truth that users often judge a system by its interface rather than its functionality. The user interface serves as the means through which users will communicate and access the system functionalities. This means that proper attention must be taken at the interface design since it is an important aspect of the system. One of the key goals of interface design is to encourage the usability of the system by ensuring that users can carry out their tasks effectively, and efficiently.

What a good User Interface Design offers:

- A quality user interface design helps make the interaction with the system as friendly and efficiently as possible in terms of accomplishing user goals.
- A poorly designed interface can cause a user to make catastrophic errors
- A Poor user interface design is the reason why so many software systems are never used

The graphical user interface type is used in designing the interface for this project.

Users of the system can navigate through the interface by using pointing device such as the mouse, or the TAB key and arrow keys on the keyboard to perform specific tasks. A standardized template was designed to ensure consistency in the layout of the web site. The header content and footer remain the same throughout the entire system. There are menu items (links) which form the basis for navigating to other pages of the web site. A user can access the site when not logged in, but upon further browsing to start commercial process, the user will be required to either log in if already registered or register to continue. The sections below will show some of the user interfaces of the TV Commercial System along with screenshots.

- **Log in interface**

This interface is designed to authenticate login credentials. The user will then have to input the required credentials in order to get access to the parts of the system the user have permissions to. This is shown in the screen dump below



- **Add customer interface**

This interface is designed to accept customer registration information. The customer must provide all the required information before he/she will be allowed to proceed to access other system functionalities.



- **Commercial rate interface**

This interface is designed to display all the available commercial types and their prices.



IV. CODING

This section of the document provides source code that will turn the design specifications into efficient, maintainable program codes. This is to create a program that exhibits a certain behavior. The coding standards and guidelines are discussed here. This stage of the project produces all the codes needed for each component of the TV Commercial System to function effectively.

A. Coding Standards

The best applications are the ones which are coded properly. This means that the code not only does its job well, but is also easy to add to, maintain and debug. According to (Mytton, 2004) looking through unfamiliar code is much easier if it is laid out well and everything is neatly commented with details that explain any complicated constructs and the reasoning behind them. Such a style might include the conventions we use to name variables and functions and how we comment our work. Any style should ensure that code can be read easily.

Since the developer of the system (John Nana Addo Bordoh) may not necessarily be the one to maintain the system, some coding standards and guidelines were taken into consideration. In this project, I chose to use the coding standard which is very flexible. Some of the standards and guidelines considered in developing the TV Commercial system include:

- Formatting and Indentation
- Inline Comments
- Classes, Subroutines, Functions, and Methods
- Source Files
- Variable Names
- Use of Braces

Because it is possible that codes be developed by several different programmers, it is paramount that all programmers should follow a well-defined style of coding in order to achieve consistency and ease of understanding. According to (Omaxy, 2012) Adherence to a coding standard has the following benefits:

- Improve readability by ensuring a “common look and feel” to the code regardless of how many people have worked on it. community gained from the past five years of language use
- improve understandability by ensuring that basic documentation is always present
- improve maintainability by improving the predictability of the code
- It gives a uniform appearance to the codes written by different engineers.
- Consistency is maintained throughout the program.
- It makes the code easier to understand for debugging and modification.
- It encourages good programming practices.
- It improves productivity.

Due to the drag and drop nature of the CASE tool (Dreamweaver), getting the structure and certain aspects of the web page come with a lot of ease. The system was coded using Hyper Text Markup Language (HTML), Hypertext Preprocessor, and Cascading Style Sheet. To makes sure that the entire project (TV Commercial System) is coded in a consistent style, no single part was written differently by different programmers. Not only does this solution make

the code easier to understand, it also ensures that any developer who looks at the code will know what to expect throughout the entire application.

B. Coding Guidelines

General coding guidelines provide a set of best practices which can be used to make programs easier to read and maintain. (Mytton, 2004) States that unlike the coding standards, the use of these guidelines is not mandatory. However, the programmer is encouraged to review them and attempt to incorporate them into his/her programming style where appropriate. In formulating the ideal coding style for this project, these guidelines by (Mytton, 2004) were taken into consideration:

- separate blocks of code into 'paragraphs' so that different sections are easily defined
- use indentation to show where control structures (if, else, while and other loops) begin and end, and where the code within them is
- variable naming conventions should be consistent throughout the code and they briefly describe that data that they contain
- Functions are named in accordance with what they do

Screenshot of Some Selected Codes

Add customer

A screenshot of a code editor showing C++ code for adding a customer. The code includes a header file, a struct for customer data, and a function named 'AddCustomer' that prompts the user for name, address, phone number, and email, then inserts the data into a database.

Login:

A screenshot of a code editor showing C++ code for user login. The code includes a header file, a struct for user data, and a function named 'Login' that prompts the user for name and password, then checks the credentials against a database.

V. TESTING

This section of the document records testing results of modules. The main aim here is to find errors. The software product is tested to ensure that it meets the business and technical requirements that guides its design and works as expected. Software testing according to (malomatia-Blog, 2012) 'is the process of validating and verifying that a computer program, application or product meets the requirements that guided its design and development, works as expected, can be implemented with the same characteristics, and satisfies the needs of stakeholders'

Testing approaches will be conducted by the development team and some users. The lists of items that will be tested include the log in section, customer section. Other sections include the customer form, employee form, product form and all the form required to be tested to ensure they work as expected.

A. Benefits of Testing

The TV Commercial System was tested, by operating it under controlled conditions to achieve the following benefits as outlined by (Kearney, 2013)

- Ensuring maximum value is obtained from the testing process
- reduced risk of failures (or incidents) once systems are transferred to live operation
- demonstrative proof that business requirements have been met

- assurance that the system will function appropriately with existing legacy systems where required and will integrate with other systems as necessary
- Assurance that the users for which the solution was designed are able to operate productively.

i. Testing Levels

Software testing levels can be grouped in various categories depending on where it was implemented in the system development process. Some of these test levels include unit testing, integration testing, system testing, conformance testing, compatibility testing, performance testing, stability testing, usability testing, security testing, acceptance testing etc.

ii. Unit Testing

Unit testing involves testing of individual software components or modules. A unit is a single, coherent piece of code. All the individual modules or components in the TV Commercial System were tested independently to determine if they function as expected. Various test cases were developed including valid and invalid credentials. By comparing the actual outcome of the test to the expected outcome, the module's effectiveness was then determined.

iii. Integration Testing

According to (softwaretestinghelp, 2007) integration testing involves testing of integrated modules or components to verify combined functionalities. Modules are typically code modules, individual applications, and client and server applications on a network, etc. Individual tested components of the system were combined into groups of components and then tested them until the system has been fully been put together as a whole.

iv. System Testing

System testing uses use case testing because the system use cases describe completely what the system must be able to do. Having designed tests for each use case, tests were conducted to check whether the use cases work together in parallel. According to (Aptest, 2013), system testing is conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements. The whole system or major subsystems are tested to determine conformance with the requirements specification of the system. System testing falls within the scope of black box testing, and as such, should require no knowledge of the inner design of the code or logic.

v. Compatibility Testing

This type of testing ensures compatibility of an application or Web site with different browsers, OSs, and hardware platforms. Because the system is a web application and is assessed by users through their web browsers, a test was carried out to check how the system operates when different web browsers are used to assess it.

This test was deemed necessary because firstly, many users may have different preferences in the use of browsers and secondly, different browsers have different configurations and settings, thus the effect of this on the system's performance should be determined. Regardless of the web browser used, the system performed as expected and delivered all its functionalities efficiently.

vi. Performance Testing

These are tests run to ensure the system developed is capable of handling large amount of data. Performance testing is performed to validate nonfunctional requirement such as software responsive time and throughput with particular workload. Performance testing also helps to identify any bottleneck that would prevent from achieving the performance goals of the system. It also helps us to find out how responsive and stable the software is when subjected to increased usage or at peak workload. Performance testing helps us to validate quality attributes like reliability, scalability, and efficiency in using the available hardware and software resources (Rao, 2012).

vii. Interface Testing (Graphical User Interface)

The Graphical User Interface test is run with the aim of determining that the software meets the requirements as mentioned in the detailed designed documents. For e.g. checking the length and capacity of the input fields provided on the form, type of input field provided, e.g. some of the form fields can be displayed as dropdown box or a set of radio buttons. So GUI testing ensures GUI elements of the software are as per approved GUI mockups, detailed design documents and functional requirements. Most of the functional test automation tools work on GUI capture and playback capabilities. This makes script recording faster at the same time and it increases the effort on script maintenance (Rao, 2012).

viii. Security Testing

According to (Rao, 2012), Security Testing involves tests to determine how the software application protects data, restricts unauthorized access and maintains functionality as intended. Security Testing is more crucial for client server and web applications.

A. Testing Approaches Adopted

The testing approach that is employed for this project includes;

- A unit testing of the various components of the system. For instance the log in component will be tested to check authentication. Various forms will be tempted to ensure domain, entity and referential integrity of the system is intact.
- There will be an integration testing to test how the various component works when they are merged together.
- There will also be a compatibility testing to test some available browsers in use today.

i. Test Case for a Single Unit (Log in Form)

In testing the log in component, the system was tested by entering nothing into the username, and password fields. The second test was putting correct entries in username field but wrong entries in the password field. The third test was putting correct credentials in both the username and password fields to check if the system works.

Test #	Inputs	Test data	Expected Outcome	Actual Outcome	Comments
1	Wrong Username and/or wrong password	Username = abcd Password = ghana	Deny access	Deny access	Testing for Wrong credentials (username and password)
2	Correct username, correct password	User name= nana Password = addo	Successful logon	Successful logon	Testing for correct Credentials (username and password)

ii. Screen Shot Confirm the Test Case (Log In Form)

Login Authentication test (Wrong username and/or password)



Login Authentication test (correct credentials)



Test evidence



iii. Test Case Enforcing Domain Integrity (Add Customer)

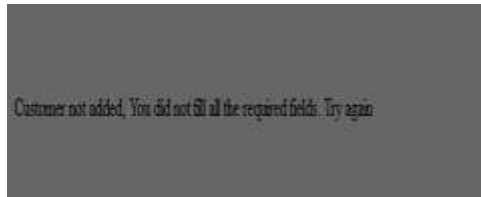
All fields are expected to be filled with the correct data type. Customer will not be added if even one required field is left unfilled.

Test #	Inputs	Expected Outcome	Actual Outcome	Comments
1	Entering data into the required fields and leaving email field empty	Prompt user to fill all the required filled	Prompt user to fill all the required filled	Testing for filling customer form that is left blank
2	Entering the required values into all fields	Confirm save	Customer is added successfully	Testing proved successful

Screen shot For Domain Integrity (Add Customer Form)
Domain integrity testing (leaving one field blank)



Test evidence



Domain integrity testing (all fields filled)



Test evidence



CustomerId	Name	Telephone	Email	Location	City
1	First National Bank	25221800	fn@fn.com.gh	Chiah House	Takoradi
2	Ghana Trade Union	418988899	tr@gtu.com.gh	Opposing Road	Accra
3	Niger SDA Hospital	252521802	nig@nhs.com	Timber Bar	Takoradi
4	Zenith bank Ghana Ltd	924466888	zen@zen.com.gh	Market Circle	Takoradi
5	Tiga Co. Ltd	48958810	tr@tiga.com.gh	Hove Cottage	Sunyasi
6	Abinga Associates	88881240	abinga@abinga.com	high way, near kings building	Sunyasi
7	Heors Co. LM	252521802	heors24@hotmail.com	hawakstee taxi station	Accra
8	MTH Telecom Ltd	418988899	mt@mt.com.gh	Adansi, Garden round	Kumasi
9	Korte Association	418988899	korte@gmail.com	bonnie park	Kumasi
10	Electoral Commission of Ghana	88881240	ec@ecg.com.gh	Habasa Ayem	Takoradi
11	Shoppers Council	48958810	shop@gmail.com	Habroo Area	Takoradi
12	GH Die Ghana Ltd	24478451	gh@die.com.gh	graphic road	Accra
13	PMC Ghana Ltd	24478451	pmcghana@mpc.com	Graphic road	Accra

iv. Compatibility test case for (bookings)

Different browsers (Internet Explorer and Mozilla Firefox) were used in testing the browser compatibility of the system.

Test #	Inputs	Expected Outcome	Actual Outcome	Comments
1	Internet explorer	To display all customer bookings	Customer bookings displayed	Testing for displaying booking details
2	Mozilla Firefox	present and/or accept booking details	present and/or accept booking details	Testing for displaying bookings form and adding booking details

Compatibility test (Mozilla Firefox) add commercial bookings

Test evidence

Booking successfully submitted. Your Secret booking ID is: 6
 Please keep your booking id safe.

BookingID	ContractID	ComType	NumberOfPlay	Time	EventID
1	1	Promotional Advert	4	Afternoon	1
2	3	Social Advert	7	Early Morning	1
3	2	Commercial	10	Evening	0
4	2	Crowd	10	Night	2
5	3	Promotional	5	Late Morning	2
6	3	Commercial	15	Afternoon	1

Compatibility test (Internet explorer) view commercial bookings

VI. TECHNICAL OVERVIEW OF THE PROJECT

This report is documentation for a TV Commercial System for Maranatha (GH) Ltd. The report is sectioned into six main parts undertaken as part of the Bsc. Business Computing and Information System course. The project

commenced with research into the core activities of TV stations and the difficulty of managing their activities using a manual system. The information gathered was used to produce the project proposal.

The proposal showed an analysis of the current system (manual system) and the problems faced. It also discussed the solutions the proposed system (TV Commercial System) will offer.

Chapter 1 is the Introduction section which gives a general overview of the project. It gives an insight into the background of the project, overview of the project environment, statement of the problem, project aims and objectives, project approach, project risks and project deliverables.

Chapter 2 talks about the specifications of the project. The analysis of the system in terms of functional and non-functional requirements was examined in this section. The main business functions of the system were identified and modeled using use cases. Other non-functional requirements such as performance, safety and security issues were dealt with. The section closes by briefly examining the class diagram of the system that was further developed in latter stages of the development.

Chapter 3: Design, the project then entered the design phase of development where the system architecture, data structure and database were modeled.

The architectural design for the system detailed deployment view (deployment diagram), interaction view (sequence diagram), and use case view (use case diagram). This phase basically described how the system will be developed to meet all the requirements specified.

The user interfaces that will be used by users to assess the system were also designed. Here some screen shots of the interfaces in the system were displayed.

Chapter 4, the system then entered into the coding phase, the system was actually made to work by creating source codes. The various modules of the system were created by coding. This was done by adhering to a number of web and coding standards and guidelines.

In Chapter 5: Testing, this chapter began with an introduction on testing followed by a general overview of the software testing methods (black box and white box). It then gave a list of some test item that will be done. After a brief description on some testing level (unit testing, performance testing, system testing, and integration testing), the chapter ended with some test case performed on some modules and then the summary for the chapter.

A. Problems Faced

Some drawbacks were encountered during this project. These include the following.

- User's unwillingness to participate during interviews.
- Frequent light out in my area
- Failures in appointments schedule with my client. There were times when meetings had to be postponed due to changes in plans of my client. This affected my overall timeline as some activities took more time to complete than expected.

B. Further Development

Certain aspects of the project which could be added to enhance the system were not realized due to time limitations and other factors which had an adverse effect on the development. Additional Functionalities that can enhance the overall system include

- **Automatic notifications sent to customers:** Extra functionality can be added to enable customers to be automatically notified when their contract ends.
- **Uploading product material:** Extra functionality can be added to enable customers to upload their physical product material irrespective of their location without the interventions of any employee.
- **Customer bills:** Extra functionality can be added to allow customer bills to be sent to their e-mails.

C. Lessons Learnt

There are series of lessons that are learnt out of every project but the distinctive experience acquired out of this project has taught me much more lessons. This project has given me a taste of the challenges of handling a real-world project. Undertaking this project has given me some knowledge which will improve my skills both as a developer and a good project manager. Although the implementation of some of the features may not be the best possible way of doing things, the project presented me with the challenge to research further into implementing Commercial Systems.

We have learnt a lot from the unique nature of software development, in the sense that it requires the use of a full project approach with the adequate management style. we have also learnt how to produce a requirements specification and design specification documents

Finally, we have been filled with the skills of time and user management since we had to deals with users mostly on scheduled date and time. We cannot conclude without recognizing the massive improvement we have gained on my communication skills by carrying out the various facts finding techniques.

Bibliography

1. Aptest. (2013, May 1). Types of Software Testing. Retrieved June 19, 2013, from <http://www.aptest.com: http://www.aptest.com/testtypes.html>
2. Association of Modern Technologies Professionals . (2013). software-development-methodologies. Retrieved Junw 21, 2013, from <http://www.itinfo.am: http://www.itinfo.am/eng/software-development-methodologies/>
3. Connolly, T., & Begg, C. (2005). Database Systems - A Practical Approach to Design,. Harlow: Pearson Education Limited.
4. Franci, B. T. (2007, September). Design a Better Database with a Data Dictionary. Retrieved July 3, 2013, from www.southbysouthwest.com: http://www.southbysouthwest.com/Dictionary.pdf
5. IBM. (2004). Designing a software application using models. Retrieved July 3, 2013, from <http://publib.boulder.ibm.com: http://publib.boulder.ibm.com/infocenter/rsdvhlp/v6r0m1/index.jsp?topic=%2Fcom.ibm.xtools.modeler.doc%2Ftopics%2Fcdp.html>
6. Kearney, A. (2013). Overview of Software Testing. Retrieved June 19, 2013, from <http://www.bcs.org: http://www.bcs.org/content/conwebdoc/7942>
7. malomatia-Blog. (2012, December 12). Software-Testing---Why-should-we-test. Retrieved June 19, 2013, from <http://www.malomatia.com: http://www.malomatia.com/English/malomatia-Blog/December-2012/Software-Testing---Why-should-we-test-.aspx>
8. msdn.microsoft.com. (2013). Retrieved July 2, 2013, from <http://msdn.microsoft.com/en-us/library/ee658094.aspx: http://msdn.microsoft.com/en-us/library/ee658094.aspx>
9. Mytton, D. (2004, September 23). why you need coding-standards. Retrieved June 18, 2013, from <http://www.sitepoint.com: http://www.sitepoint.com/coding-standards/>
10. Omaxy. (2012). Coding Standards & Best Practices . Retrieved June 18, 2013, from <http://www.omaxy.com: http://www.omaxy.com/development/developer-guidelines/coding-standards/>
11. Pressman, R. S. (2005). Software Engineering A Practioners Approach ((6th Edition ed.) ed.). New York: McGraw-Hill Inc.
12. Rao, J. (2012, July 26). all-types-of-software-testing. Retrieved June 19, 2013, from <http://www.softwaretestingsoftware.com: http://www.softwaretestingsoftware.com/all-types-of-software-testing/>
13. Rouse, M. (2006, February). definition/data-structure. Retrieved June 20, 2013, from <http://searchsqlserver.techtarget.com: http://searchsqlserver.techtarget.com/definition/data-structure>
14. Shah, B. (2008, April 11). Fact Finding Techniques. Retrieved June 21, 2013, from <http://systemanalysisanddesign.blogspot.com: http://systemanalysisanddesign.blogspot.com/2008/11/fact-finding-techniques.html>
15. softwaretestinghelp. (2007). types-of-software-testing/. Retrieved June 19, 2013, from <http://www.softwaretestinghelp.com: http://www.softwaretestinghelp.com/types-of-software-testing/>
16. Sommerville, I. (2004). Software Engineering. <http://ifs.host.cs.st-andrews.ac.uk/Books/SE7/Presentations/PDF/ch11.pdf> , 10.
17. Sommerville, I. (2007). Software Engineering. Manchester: Pearson Education Ltd.
18. Taflinger, R. F. (1996, May 28). A Definition of Advertising. Retrieved October 8, 2012, from wsu website: <http://public.wsu.edu/~taflinge/addefine.html>
19. Weaver, P. (2004). Success in Your Project. Harlow: Pearson Education Limited.
20. Williams, R. (2006). Real-Time Systems Development. Oxford: Elsevier.