SOFTWARE DESIGN

DESCRIPTION

# Overview

## Scope

The developed product is a system that implement a web-app that provides the act of online business. Objective of the system is to make the process of online selling and buying easier. Each design concern of the stakeholders are topic of at least one design view and these design views are described with corresponding design elements and modeled with related UML diagrams. The document is prepared in IEEE 1016-2009 standards.

# System Architecture

The software will consist of a database, an Operating System, a web server, VB Code, and several APIs for third parties.



## Purpose

The purpose of this document is to provide a description of the design of the web-app to allow for software design to proceed with a perceptive of the design that is to be structured and how the process of it develops. The topics of, general description of design elements and their interactions, how the system will be structured, data & functional structure are to be further discussed in order to help producing test cases, and help in maintenance services, and also satisfy requirements, design details indicated in the SRS document.

## Intended audience

The intended audience of this document is all major stakeholders which include the development team , the project owner, the project mentor ), faculty members and anyone evaluating the project.

## References

1. IEEE. IEEE Std 1016-2009 IEEE Standard for Information Technology – System Design –

Software Design Descriptions. IEEE Computer Society, 2009

1. Software Specification Requirements (SRS) Document of Human Body Health Status Software prepared by

# Definitions, Acronyms and Abbreviations

All the definitions, acronyms and abbreviations which are used in this document are described in the following table.

|  |  |
| --- | --- |
| Block diagram | A diagram showing in schematic form the general arrangement of the parts or components of a complex system or process. |
| Class Diagram | A type of static structure diagram in UML that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes |
| DDB | Distributed Database |
| HTTPs | Hypertext Transfer Protocol Secure |

3

|  |  |
| --- | --- |
| IDE | Integrated Development Environment |
| IEEE Standard | International Electric Electronic Engineering Standards 1016-2009 |
| PC | Personal Computer |
| SQL | Structured Query Language |
| SDD | Software Design Description |
| SDK | Software Development Kit |
| SRS | Software Requirement Specification |
| Stakeholders | Any person with an interest in the system who is not a developer of the  system. |
| StarUML | Design tool of diagrams |
| State Transition  Diagram | A type of static structure diagram in UML that describes the transition of the system functions |
| UI | User Interface |
| Use Case  Diagram | A type of static structure diagram in UML that describes user's interaction with the system |
| User | Person who wants to use the system |
| User Interface | An interface that our system contacts with the user of the system. It gets  all needed information for its running, from user to our system. |
| XML | Extensible Markup Language |

# Conceptual Model for Software Design Descriptions

This section includes basic e commerce operations and structure, concepts and context of SDD in which the documentation is prepared. The purpose of conceptual model is to give a better understanding of system terminology and software life cycle that the system resides on. The conceptual model also gives information about stakeholders who will use SDD and how the SDD will be used.

4

## Software Design in Context

The Ecommerce system will be designed as a web application. The system will be implemented with HTML, CSS, Angular JS, JavaSpringBoot using Visual Studio Code as IDE using Laravel framework. The system will try to give the most accurate result in a most applicable, proper and correct time; so, it can respond to users’ wants correctly and quickly. The speed of the system depends on the computational specifications of the search engine, browser, and the computer.

Our Ecommerce System is planned to be an application that run on the web browser in any personal computers which have access to Internet. The system can be accessible to all users who have some web browsers installed in their Linux, windows, mac or any other operating systems.

## Software Design Descriptions within the Life Cycle

This software will be created following IEEE standards. The primary milestones of this system are requirements analysis, design description analysis, implementation and finally verification and validation.

### Influences on Software Life Cycle Products

Firstly, all interfaces should be designed. Before officially launching the system, user interface should be shown with sample examples to the stakeholders. As a result of this process, stakeholders can share their ideas and requirements about the Ecommerce System. Finally, the system can be published as fully functioning website.

Furthermore, SDD will guide us all the way through the system. According to this document or the first phase, some requirements can be added or removed from the software requirements. Consequently, requirements of the stakeholders can be met more precisely after each sprint of our development process.

5

### Design Verification and Design Role in Validation

Verification is the process that we will test System whether it meets a set of design specifications. In this process, we will look the SRS and SDD documents for correctness of specifications. We will control that whether all functional and nonfunctional requirements are correctly implemented according to the requirements of SRS and SDD documents. Furthermore, we will control that whether the design viewpoints of the final Human Body Health Status System are met in the viewpoints part of the SDD document.

Validation is the process that the stakeholders and developers decide if the Human Body Health Status System is consistent with the main goal provide the health status of a human body according to certain inputs and to suggest some exercises that can improve the current scenario of his/her physical health.

After the complete implementation of system, the testing process will be handled with SDD influenced test plans and cases.

# Design Description Information Content

## Introduction

Software design description of the Ecommerce WebApp analyzes how the system will be designed and implemented. This section investigates these according to identification of the SDD, identified design stakeholders and design concerns, related design viewpoints, design views, overlays, rationale and languages. Furthermore, this section includes design elements which are design entities, attributes, relationships and constraints.

## SDD Identification

Ecommerce WebApp will be released by the end of June 2022 after validation and verification tests. Prototype of the system will be shown in the middle of June 2022.

Modification and distribution Ecommerce System can only be done by the copyright holder who is Developer team because of the exclusive rights property.

Scope, references, context and summary can be found under the section “Overview”.

Glossary can be found under the section “Definitions, Acronyms and Abbreviations”.

6

## Design Stakeholders and Their Concerns

Stakeholders comprise a developer team and Technology Department of Computer Science Engineering members. Mainly concerns of the stakeholders are accuracy of the information which is provided by the system because it must be perfect. In addition, this project may be a little difficult to design and develop.

## Design Views

Design views help design stakeholders about focusing on design details from a specific perspective and meeting relevant requirements. Each identified design concern must be the topic of at least one design view so that SDD is complete. Each design concern identified in the previous subsection is the topic of most of the design views in this document; thus, this SDD is completed. For example, concerns about cost are a topic of composition view. Moreover, concerns about accuracy of data are a topic of logical view. In this document, context, composition, logical, dependency, information, patterns use, interface, interaction and state dynamics views will be explained in section 4.5 as their corresponding viewpoints. For some views, relevant UML diagrams will be shown in order to clarify.

## Design Viewpoints

This document describes context, composition, logical, dependency, information, patterns use, interface, structure, and interaction and state dynamics viewpoints.

***Context Viewpoint:*** It describes the relationships, dependencies and interactions between the system and its environment such as users and other interacting stakeholders. Interactions between the system and its actors are very intense, hence concerns of this viewpoint are important and suitable for the Human Body Health Status System. It includes a use case, context and block diagram showing the system boundary.

***Composition Viewpoint:*** It describes how the design subject split up into its components and which roles these components have. It can be used in estimating cost, staffing and scheduling duties of a development team. It includes a deployment and component diagram.

***Logical Viewpoint:*** It describes class structures, interactions between them and how they are designed and implemented. Also, it supports development and reuse of existing logical components. It includes a class diagram which defines objects and classes, and relationships between them.

7

***Dependency Viewpoint:*** It describes the components of the system and dependencies between these components. It gives information about shared information and order of execution of these components.

***Information Viewpoint:*** It describes data items, data types and classes, data stores and access mechanisms. It gives information about data attributes.

***Patterns Use Viewpoint:*** It describes design patterns and usage of design patterns which meet design ideas of the project.

***Interface Viewpoint:*** It describes the details of external and internal interfaces. It provides information to the designers, programmers and testers before proceeding with the detailed design of the system. This also provides designers, programmers and testers to use the system as a random user.

***Interaction Viewpoint:*** It describes the sequence of actions and how, why, where and at what level actions occur in the system. It is preferred to use state dynamics views in detail for this project.

***State Dynamics Viewpoint:*** It describes the internal behavior of the system. System dynamics include modes, states, transitions and reactions to events. It gives information step by step about the system operation. It includes a state machine diagram which defines conditions, states, transitions and relationships between them.

## Design Elements

Any item which appears in a design view is named as design elements. It may be one or some of these subcases; design entity, design relationship, design attribute and design constraints. All design elements are defined with subcases under their corresponding viewpoint in section 5 of the software design description.

## Design Overlays

Design overlays are usually used to add information to the existing views. This concept will be explained clearly, when necessary, in the design viewpoints section which is 5.

## Design Rationale

The Object-Oriented approach was chosen while designing because by this way the hardware part and software part will be combined easily. Software part includes parsing and reading notes and transmitting data. To design these lots of packages were used. These packages are connected

8

between each other and they can be controlled separately. Furthermore, for the hardware part a package was used and there is another package to combine software and hardware parts.

## Design Languages

In this document Unified Modeling Language (UML) will be used as the modeling language for the diagrams. The modeling language is used for emphasizing the static structure and dynamic behavior of the system.

# Design Viewpoints

## Introduction

This section provides several main design viewpoints of Ecommerce System with their corresponding design concerns and appropriate design languages.

Respectively, context, composition, logical, dependency, information, patterns, interface, structure and interaction viewpoints are defined in the following subsections. Short descriptions relating a minimal set of design entities, design relationships, design entity attributes, and design constraints are provided for each viewpoint.

## Context viewpoint

The context viewpoint of Ecommerce WebApp software shows the functions provided by a design subject with reference to an explicit context. The services are the functions, which describes the relationships, dependencies, and interactions between the system and its environment like users and other stakeholders.

### Design concerns

Design concerns consist of services, actors and system boundaries. Ecommerce WebApp is formed of a website interface. There are two type of actors who uses the system; people who access the system regularly (users with an account) and who access it infrequently (guest Users). Therefore, the constraints vary according to users.

The user uses the Human Body Health Status Software interface to access the calculator login or signup page. Information flow between the user and the Human Body Health Status System is provided by the interface. Below diagram shows the system boundary which includes the relationship between the user and the other major components of the system.

9

A diagram of a flowchart

Description automatically generated

**Figure 2: Activity Diagram For Customer**

Figure 3 given below describes the activity done by the customer within the system

A diagram of a diagram

Description automatically generated+

**Figure 3: Use Case Diagram of the** Customer and Admin

11

### Example Languages

The diagrams given in the previous subsections are created by the UML. One of these diagrams is the block diagram describing interrelationships of a [system.](http://en.wikipedia.org/wiki/System) Another one is the context diagram defining the boundary between the [system](http://en.wikipedia.org/wiki/System) and its environment. The last one of them is the use case diagram showing user interactions with the system.

## Composition Viewpoint

The purpose of the composition viewpoint of Recommendation System is to define the system as a composition of its subsystems. The project is formed by 4 main submodules: GUI, Calculator, Database and Account Manager. Detailed explanation about the relations between these modules will be explained in coming sections.

12

### Design Elements

The project is formed of interfaces and classes inside packages as design entities. There are three packages which are User Interface, Calculator and Account Manager which allow the system to be run as an application. A figure showing the relations between these modules is given below:

## Logical Viewpoint

The purpose of the Logical viewpoint is to elaborate existing and designed types and their implementations as classes and interfaces with their structural static relationships. For each entity, there will be a diagram to overview the entity and then a table that name, return type; visibility of the entity/class diagram is shown in. Also, definitions of each element is provided.

After all elements are explained, the class diagram that shows relationships between the classes is drawn.

13

### Design Concerns

The logical viewpoint is employed to show the development and reuse of abstractions and their implementations. This means, object-oriented programming simplifies to maintain and modify existing code while new objects are created. Since identifying object classes is often a difficult part of object-oriented design, during the implementation phase of the project there can be some changes in object identification.

### Design Elements

The project has three packages named User Interface, Calculator and Account Manager. All packages have different classes. All package connections and their classes can be seen in the figure:

A diagram of a software application

Description automatically generated with medium confidence

**Figure 6: Class Diagram of Ecommerce System**

14

### Example Languages

A class diagram which describes the structure of a system by showing classes of the system has been given in the previous sections by using UML modeling language.

## Dependency Viewpoint

The Dependency viewpoint specifies the relationships of interconnection and access among entities. These relationships include shared information, order of execution, or parameterization of interfaces.

### Design Concerns

Dependency viewpoint helps maintainers to isolate entities causing system failures or resource bottlenecks. In producing the system integration plan, the system is identified with the sub-modules and the components which are dependent on each other.

### Design elements

The product is composed of the user interface and classes inside packages as design entities. Detailed explanation about the dependency between the entities and the modules are explained with the component and deployment diagrams in the section 5.3.2.

### Example Languages

The component diagram showing functional (logical) decomposition of the system in the section 5.3.2 and the package diagram depicting the [dependencies](http://en.wikipedia.org/wiki/Dependency_(UML)) between the [packages](http://en.wikipedia.org/wiki/Package_(UML)) in the section 5.9.2 are created by UML.

## Information Viewpoint

This section is not required for our project.

## Patterns Use Viewpoint

This section is not required for our project.

## Interface Viewpoint

In this part of the document, the details of external and internal interfaces will be defined. There shall be two interfaces in Human Body Health Status System, which are Database Interface and user interface. Database interface is hidden from users. Clients interact through user interface and

15

provide necessary information’s for calculation procedures and those information’s are stored in the database.

.

### Design Concerns

Interface viewpoint provides information to the designers, developers and testers before proceeding with the detailed design of the system. It also informs them about how cooperating entities will interact. With the ease of each interface descriptions, designers and developers can know internal and external connections of system to develop it. This contributes ease of maintenance.

### Design Elements

In this subsection, user interface of the home page of the Ecommerce WebApp System is shown. In addition, usage of these interfaces is given in details.

A screenshot of a computer

Description automatically generated

**Figure 7: User Interface of Human Body Health Status System**

Figure 7 shows the only interface of the Human Body Health Status System. This interface shows what a client sees when they first visit the website. Two options were provided at this page; one for using the basic website features without creating any account (used by temporary users) and the other for using all available features after creating a valid account with necessary information (used by regular users). Users can choose between these two options based on their

16

preferences. Users who selected the first option will be eligible two try out some services provided by the website so that they can get a small clear-cut understanding of the services provided by the Human Body Health Status System. The users who selected the second option will be eligible for trying out all the services and functionalities provided by the Human Body Health Status System without any restrictions.

All necessary UML component diagrams have been given in the previous sections of this SDD document.

Developer manages interaction between parts.

Detailed component diagram is given in section 5.3 and class diagram given in section 5.4.

### Example languages

UML package diagram showing interaction between the parts of the system and the user has been given in the previous sections by using UML modeling language.

## Interaction viewpoint

In this section, main functionalities of the system are given by the help of sequence diagrams. Moreover, it defines strategies for interaction among entities.

### Design Concerns

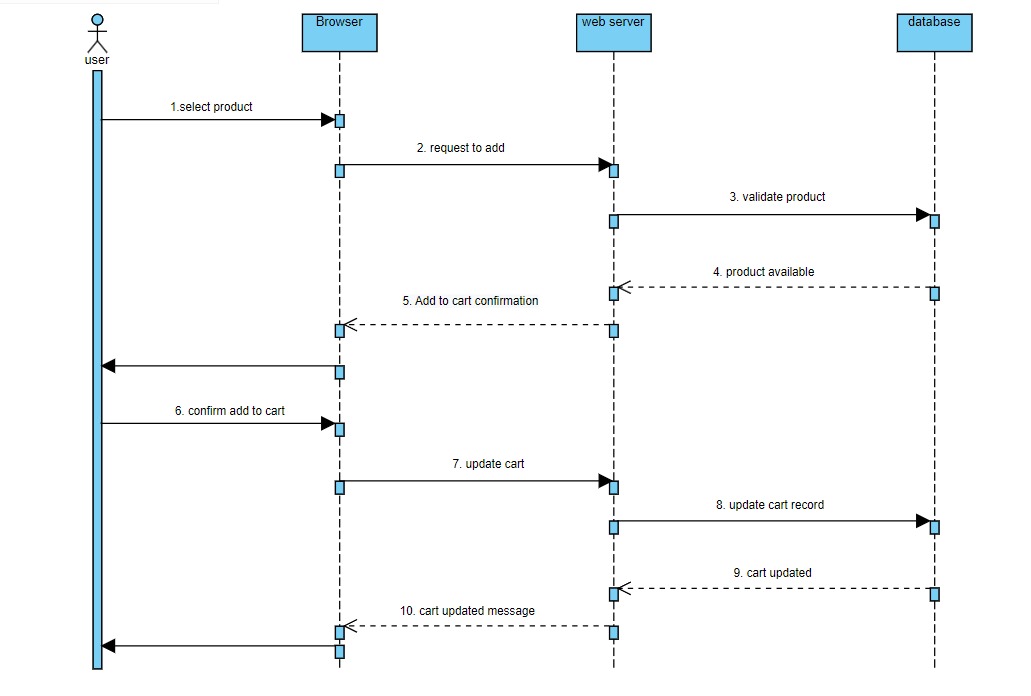
For designers, this section includes evaluating allocation of responsibilities in collaborations and description of interactions in terms of messages among affected objects in fulfilling required actions with the help of sequence diagrams.

### Design Elements

Given sub-sections below, user and system interactions are illustrated.

* + - 1. ***Interactions of Regular Users***

18



**Figure 9: Sequence Diagram for Adding to cart of Regular Users**

Above sequence diagram shows the different ways in which a regular user interacts with the system. They have basically two courses of action: Sign-up or Login. The sequence of steps and related actions in both these alternatives are shown in the sequence diagram.

* + - 1. ***Interactions of Temporary Users***

19

A diagram of a diagram

Description automatically generated

**Figure 10: Sequence Diagram for Showing Products for user**

The sequence diagram above shows the interactions between the user and system to show the user the products .

### Examples Languages

Sequence diagrams which show how processes operate with one another and in what order has been given in the previous section by using UML modeling language.

20

21