# **HANOI UNIVERSITY Faculty of Information Technology (FIT)**

**SOFTWARE QUALITY ASSURANCE**

**FINAL REPORT**

**Faculty**: **Information Technology**

**Module Name**: **System Analysis & Design**

**Year**: **Fall 2020**

**Topic**: **Pet Care Center Management System**

**Group: 3**

**Group Members**: **Ly Trung Kien**

**Le Duc Long**

**Ton Nu Tu Anh**

**Nghiem Thi Xuan Thuy**

**Dang Tran Trung Hieu**

**Group 5 – SQA Tutorial 02**

**TABLE OF CONTENTS**

GROUP ROLES..............................................................................................................................3 1. Business Modeling ……………………......................................................................................4

2. Requirements Definition…………………... ..............................................................................4

3. Mockups &Prototype …………………......................................................................................4

4. Analysis & Design…………………...........................................................................................4

5. Testing….. .................................................................................................................................. 7

6. Demo video……………………………………………………………………………………7

7. Conclusion……………………………………………………………………………………..7

**GROUP ROLES**

|  |  |
| --- | --- |
| **Members** | **Roles** |
| Ly Trung Kien | Project Manager, Business Modeling |
| Le Duc Long | Project Developer, Database Designer, Tester |
| Ton Nu Tu Anh | Business Modeling, UI designer |
| Nghiem Thi Xuan Thuy | Designer |
| Dang Tran Trung Hieu | Requirement Analysis |

1. **Business Modeling (Ly Trung Kien, Ton Nu Tu Anh)**

Tu Anh is the business designer and Kien is the business process analyst. Tu Anh is responsible for the business architecture. She details the specification of a part of the organization by describing the workflow of one or several business use cases, along with defining the responsibilities, operations, attributes, and relationships of one or several business workers and business entities. Meanwhile, Kien leads and coordinates business use-case modeling by outlining and delimiting the organization being modeled; for example, establishing what business actors and business use cases exist and how they interact.

1. **Requirements Definition (Dang Tran Trung Hieu, Nghiem Thi Xuan Thuy)**

Hieu and Thuy are requirements specifiers. They detailed the specification of a part of the system's functionality by describing the Requirements aspect of one or several use cases and other supporting software requirements. They are also responsible for the use-case package, and maintain the integrity of that package. Hieu and Thuy detailed the use cases and diagrams and made them consistent with other requirements discipline artifacts. Besides, they captured requirements on the user interface, including usability requirements.

1. **Mockups & Prototype (Ton Nu Tu Anh)**

Tu Anh has implemented a prototype to demonstrate the Amazing e-commerce website functionality using Figma for the UI design for the system. The prototype is provided with a simple Graphical User Interface and has some function of the system specified in requirements definition such as sign-in, add product to cart, remove product from cart…

1. **Analysis & Design (Ly Trung Kien, Le Duc Long)**

As a software architecture, Long and Kien lead and coordinates technical activities and artifacts throughout the project. Their primary responsibility is to establish the overall structure for each architectural view: the decomposition of the view, the grouping of elements. Therefore, she has a breadth view of the system architecture. The table below describes in detail her activities as a software architecture in the Analysis & Design workflow.

|  |  |  |
| --- | --- | --- |
| **Activity** | **Description** | **Output Artifacts** |
| Prioritize Use Cases | * Define input to the selection of the set of scenarios and use cases that are to be analyzed in the current iteration. * Define the set of scenarios and use cases that represent some significant, central functionality. * Define the set of scenarios and use cases that have a substantial architectural coverage or that stress or illustrate a specific, delicate point of the architecture. | Software Architecture Document |
| Architectural Analysis | * Define a candidate architecture for the system, based on experience gained, from similar systems or in similar problem domains. * Define the architectural patterns, key mechanisms and modeling conventions for the system. * Define the reuse strategy. | Use Case Diagrams, Sequence Diagrams, Activity Diagrams |
| Identify Design Mechanisms | Refine the analysis mechanisms into design mechanisms based on the constraints imposed by the implementation environment | Design Model, Software Architecture Document |
| Incorporate Existing Design Elements | * Analyze interactions of analysis classes to find design classes. * Refine the architecture, incorporating reuse where possible. * Identify common solutions to commonly encountered design problems. * Include architecturally significant design model elements in the Logical View section of the Software Architecture Document. | Design Model, Software Architecture Document |

As designers, Kien and Long are in charge of defining the responsibilities, operations, attributes, and relationships of several classes, and determine how they will be adjusted to the implementation environment of the Computer Shop Management System. The table below describes their activities as designers in the Analysis & Design workflow.

|  |  |  |
| --- | --- | --- |
| **Activity** | **Description** | **Output** |
| Use-Case Analysis | * Identify the classes which perform a use case’s flow of events. * Distribute the use case behavior to those classes, using use-case realizations. * Identify the responsibilities, attributes and associations of the classes. * Note the usage of architectural mechanisms. | Analysis class, Use-Case Specification, Design Model, Analysis Model |
| Use-Case Design | * Refine use-case specifications in terms of interactions. * Refine requirements on the operations of design classes. | Use-Case Specification |
| Class Design | * Ensure that the class provides the behavior the use-case realizations require. * Ensure that sufficient information is provided to unambiguously implement the class. * Handle non-functional requirements related to the class. * Incorporate the design mechanisms used by the class. | Design Class |

As a database designer, Long essential obligation is defining the tables, indexes, views, constraints, and other database-specific constructs needed to store, retrieve, and delete persistent objects. The table below describes her activities as a database designer in the Analysis & Design workflow.

|  |  |  |
| --- | --- | --- |
| **Activity** | **Description** | **Output** |
| Database Design | * Ensure that persistent data is stored consistently and efficiently. * Define behavior that must be implemented in the database. | Database Script |

1. **Testing(Ly Trung Kien, Le Duc Long)**

As a tester, Kien is responsible to write unit testing by using Mocha test framework for JavaScript. He will execute all test case to make sure that our code is qualified. The output of this phase is test report which represents all test case and its result after using mocha test.

1. **Demo video** ( Le Duc Long, Ton Nu Tu Anh):

Long and Tu Anh are in charge of doing an demo video for our project. In this video, they have to represent about the project. It may be contain a script of Project Plan, code explanation, final result and evaluation of this software.

1. **Conclusion**

Based on knowledge acquired from this course and previous courses, we have designed a management system with requirements that match real world business (Amazing e-commerce Website). In our project, we have modeled business context for the system, capturing and describing system requ irements using both textual and UML notation. Moreover, we have designed the system using Model-View-Controller models as a foundation; therefore, improve robustness and reusability of system components. Also, a runnable prototype is built based on the requirement to demonstrate system functionality. Nevertheless, there is still room for improvement in the project. Because our knowledge base limit and inexperience in System Quality Assurance particularly and Information Technology in general, erroneous in our work are inevitable. Therefore, we will continue to improve our knowledge to deliver better products in the future.