Strategic Marketing Simulator

Project team number

Feasibility Study and Project Plan

Senior Project – CIS 4911

Jeffrey Carman

Javier Andrial

Joseph Cilli, Project Manager

Course Instructor: Masoud Sadjadi

Copyright stuff

Abstract

*Students of the FIU business school take marketing courses to give them a solid understanding of business activities as they relate to target markets, development of pricing models, the promotion of goods and services, the management of business relationships and consumer behavior.  Much of the curriculum is theory as it pertains to marketing concepts, but there is something lacking.  The subject application, Strategic Marketing Simulator, allows students and instructors to participate in a mockup of a hotel market, where the way a marketing budget is allocated and the state of the market has a direct impact on how the market trends during an interval of time, a period. Additionally, students gain an understanding on how political and social policy, force majeure, supply and demand etc., can affect one company's market share over another.*

Table of Contents

[1. Introduction 3](#_Toc429248592)

[1.1. Problem definition (very similar to RD). 3](#_Toc429248593)

[1.2. Design methodology used e.g., identify software process model, ease of creating a design from the systems requirements, types of models (UML models) used to represent the design. 3](#_Toc429248594)

[1.3. Definitions, acronyms, and abbreviations. 3](#_Toc429248595)

[1.4. Overview of document 3](#_Toc429248596)

[2. System Design (i.e., overall system design) 3](#_Toc429248597)

[2.1. Overview – high-level description of the system design (architecture) e.g., provides a package diagram showing the major subsystems and briefly describes each subsystem. Relate the system decomposition to the requirements of the system. Use at least two (2) architectural patterns. 4](#_Toc429248598)

[2.2. Subsystem Decomposition – provide a detailed description for each of the major subsystems. Identify the requirements associated with each subsystem. 4](#_Toc429248599)

[2.3. Hardware and Software Mapping – map subsystems to h/w and s/w. The h/w and s/w are for the systems to be implemented. May include a deployment diagram showing the associations between the subsystems and hardware. 4](#_Toc429248600)

[2.4. Persistent Data Management – identify data that needs to be stored and the structure of the data. Use a data dictionary to represent the initial data extracted from the use cases. 4](#_Toc429248601)

[2.5. Security/Privacy – describe user authentication processes, encryption of data, and use of firewalls or security servers. 4](#_Toc429248602)

[3. Detailed Design 4](#_Toc429248603)

[Introduce the detailed design chapter (one or two paragraphs) 4](#_Toc429248604)

[3.1. Overview – briefly describe the behavior and structure of each subsystem. 4](#_Toc429248605)

[3.2. Static model – detailed description of the structure for each subsystem. May include detailed class diagrams. Place diagrams (e.g., minimal class diagram, detailed class diagram per subsystem) inline. Use at least four (4) design patterns. 4](#_Toc429248606)

[3.3. Dynamic model – state machine diagram for the main control object in each subsystem. Include the design of the main algorithms used in the problem solution. Refinement of the sequence diagram from the analysis model. Place diagrams inline. 4](#_Toc429248607)

[3.4. Code Specification - describe the class interfaces (attributes and method signatures) and constraint (invariants, pre-condition and post-conditions) for the main control object in each system. Code should be in Appendix C. 4](#_Toc429248608)

[4. Glossary - define terms used in document, especially domain specific terms. 4](#_Toc429248609)

[5. Appendix 4](#_Toc429248610)

[5.1. Appendix A - Use case diagram for use cases being implemented. 4](#_Toc429248611)

[5.2. Appendix B - Use cases being implemented (from the RD). 4](#_Toc429248612)

[5.3. Appendix C – Documented class interfaces (code) for the subsystem(s) you will implement and the constraints. 5](#_Toc429248613)

[5.4. Appendix D - Diary of meeting and tasks. 5](#_Toc429248614)

[6. References 5](#_Toc429248615)

# Introduction

## 1.1. Problem definition (very similar to RD).

University instructors need a means for having their students run simulations of a hotel room sales market to show the impact that different marketing budget allocations will have on the market. These allocations need to take the form of dollars spent on marketing personnel, advertising, and promotions; along with type of hotel and the location of the hotel. The way a student utilizes their budget will ultimately affect the market by either giving him or her competitors a market advantage. By the same token, these students need a means for interpreting the effects of budget allocations that other students have in the same market. Finally, instructors need a way to track these results over varying intervals of time.

## 1.2. Design methodology used e.g., identify software process model, ease of creating a design from the systems requirements, types of models (UML models) used to represent the design.

The Strategic Marketing Simulator is being designed using a modified version of scrum wherein user requirements are communicated by the Product Owner to the scrum team in the form of user stories, concise descriptive sentences of what each actor on a system needs to be able to accomplish. The process is being performed in sprints, two week periods consisting of requirement elicitation and analysis, modeling, development, and testing. At the end of each sprint a delivery (sellable software product) is submitted to the product owner.

During the requirements elicitation phase of development, the product owner conveys is desired program functionality. The requirements are analyzed to produce a system design which is a set of subsystems encompassing the entire system. After having the system design, the design plan is further refined by using object models and class models to show how the classes of object oriented system will interact with one another and users of the system.

Once the modeling for the system is sufficient, development being the process of implementing the design of the system. Next, the software is tested using unit testing, integration testing, and validation testing. Once the software team is satisfied with the testing, a deliverable can be submitted for review.

## 1.3. Definitions, acronyms, and abbreviations.

## 1.4. Overview of document

# 2. System Design (i.e., overall system design)

Introduce the system decomposition chapter (one or two paragraphs).

The Strategic Marketing Simulator is composed of four subsystems.  The database subsystem is responsible for all database transactions being made by any class in the system.  The other subsystems communicate with the database to perform a variety of tasks such as adding a user record to the database or accesses a value needed for the Market subsystem.  The Market subsystem is responsible for interpreting the effect of a group’s strategic decisions on the market.  The Market subsystem communicates with the database subsystem to retrieve information about which groups are applying a certain decisions.  Once the Market Ranking subsystem has made a determination, it can then pass the values to the Database subsystem storage for later use.

The third subsystem is the Main Controller subsystem.  The Main Controller handles the game creation and keeps track of the current periods.  The Main Controller subsystem communicates with both the database subsystem and the Market subsystem such that market information stored in the database subsystem can be used in conjunction with the Market subsystem to display market results on a webpage. The email subsystem is responsible for sending emails to users upon account creation to verify their account and for password reset.

## 2.1. Overview – high-level description of the system design (architecture) e.g., provides a package diagram showing the major subsystems and briefly describes each subsystem. Relate the system decomposition to the requirements of the system. Use at least two (2) architectural patterns.

The architectural patterns used in the development of this system were three-tier and model-view-controller patterns.  These patterns were adopted because during the requirements elicitation of analysis phase of the project, apparent subsystems began to emerge that have a three-tier like pattern, which has a controller, database, and a graphic user interface.  Taking into consideration that emerging pattern and the team’s understanding of the system made the model-view-controller also relevant due to the apparent subsystems’ disconnect between between one another in functionality and requirements.

The Strategic Marketing Simulator system can be decomposed into four subsystems.  The MainController subsystem is responsible for creating games, users, and groups.  All system input and output is communicated using the MainController subsystem as an intermediate between the front end and the rest of the back end.  The second system, the database subsystem is responsible for all database transactions and the storing of game and user data.  The third subsystem is the Market subsystem which is responsible for calculating all groups’ Strategic Decisions at the end of each period, and passes values to the MainController subsystem for display on the back end or to be handed to the Database subsystem storage for later use.  The fourth subsystem is the Email subsystem, which is responsible for sending emails to users upon account creation and for password reset.

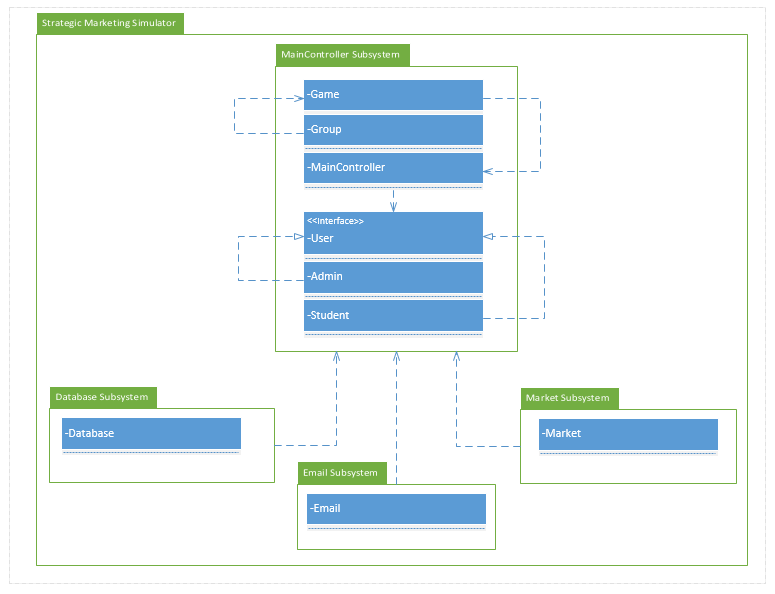


Figure 1:  Package Diagram.

## 2.2. Subsystem Decomposition – provide a detailed description for each of the major subsystems. Identify the requirements associated with each subsystem.

## 2.3. Hardware and Software Mapping – map subsystems to h/w and s/w. The h/w and s/w are for the systems to be implemented. May include a deployment diagram showing the associations between the subsystems and hardware.

## 2.4. Persistent Data Management – identify data that needs to be stored and the structure of the data. Use a data dictionary to represent the initial data extracted from the use cases.

## 2.5. Security/Privacy – describe user authentication processes, encryption of data, and use of firewalls or security servers.

# 3. Detailed Design

# Introduce the detailed design chapter (one or two paragraphs)

## 3.1. Overview – briefly describe the behavior and structure of each subsystem.

## 3.2. Static model – detailed description of the structure for each subsystem. May include detailed class diagrams. Place diagrams (e.g., minimal class diagram, detailed class diagram per subsystem) inline. Use at least four (4) design patterns.

## 3.3. Dynamic model – state machine diagram for the main control object in each subsystem. Include the design of the main algorithms used in the problem solution. Refinement of the sequence diagram from the analysis model. Place diagrams inline.

## 3.4. Code Specification - describe the class interfaces (attributes and method signatures) and constraint (invariants, pre-condition and post-conditions) for the main control object in each system. Code should be in Appendix C.

# 4. Glossary - define terms used in document, especially domain specific terms.

# 5. Appendix

## 5.1. Appendix A - Use case diagram for use cases being implemented.

## 5.2. Appendix B - Use cases being implemented (from the RD).

## 5.3. Appendix C – Documented class interfaces (code) for the subsystem(s) you will implement and the constraints.

## 5.4. Appendix D - Diary of meeting and tasks.

# 6. References