

SE2890 Software Engineering Practices

Dr. Walter Schilling

Spring, 2011-2012

On the midterm and final exam, you are permitted one 8.5 x 11 sheet of paper with notes.

The following paragraph describes a system which you have been asked to design and implement. This system will be used for several questions on the exam. Read through the system description so you are familiar with it and think about the potential design questions which can be asked.

THE MSOE Dining services is attempting to receive higher ratings from its student customers. As part of this process, several new innovations are in development that will be rolled out in the coming years. As part of this effort, the consulting firm of Dewey, Cheetham, and Howe has been retained to design and implement a new system, the “MSOE Advanced Dining Reservation System” (MADRS). The MADRS system allows the MSOE dining services to better plan for student dining plans and meal preferences.

The MADRS system allows students to reserve a dining time up to 7 days in advance based upon their preferences for Breakfast, Lunch, and Dinner. To do this, a student authenticates with the system by swiping their student ID in a card reader at a kiosk and entering their MSOE password. These entries are verified by the existing LDAP system. If a user has been blocked, a message will be displayed to the user indicating that they have been blocked by the chef and they will be denied further access to the system. Through the usage of buttons on the touch screen, the student can make a table reservation inside of the dining hall for his or her self, as well as up to three other registered MSOE students by entering their login id. The system will not allow the user to add another student who already has a dining reservation made for the given meal. The reservation system will automatically send a calendar invite to existing campus Outlook system indicating that dinner plans have been made.

Once a reservation is made, any student who has a reserved place for dinner student may “preorder” their food for dinner. To preorder food, the student again will swipe their id at a kiosk, select from their existing dining reservations, and select from the food which is available during the times that the reservation has been made. In addition to normal dining options, the system also intends to offer “gourmet” dining options, such as steak, Caviar, Smoked Salmon, Roasted Duck, and Chicken Cordon bleu. “Gourmet” options will be available on Sundays for dinner as well as during “special” occasions on campus. “Gourmet” options are available to students who receive a 3.0 or higher GPA, which will be checked through the existing Jenzebar system. When preordering food, the system will inform the students as to the cost of their dinner.

A student who has preordered their food selection may cancel their dining order up to 4 hours in advance of their dining time by using a kiosk, unless the student has ordered “gourmet foods” which require 24 hours notice to cancel. Dining table reservations also may be canceled up to 4 hours in advance by the person who initially made the reservation. Canceling the reservation will automatically send a cancelation notice to Outlook through the existing MSOE Outlook system.

The chef will use the system to determine how much gourmet food to order by printing out dining orders. The system will keep track of how many times each student has canceled dining reservations, and if this number is too high, the chef may block a user from accessing the system.

The system will automatically bill the student through the existing accounts system when the student checks in for dinner. In the event that a student has a reservation and does not show within 30 minutes of the reservation time, they will be charged double for their dining options.

1. Week 1 - Introduction

(a) Lecture 1: Introduction

- i. Recognize the economic impacts of software failure
- ii. Discuss the ramifications of software failure
- iii. Understand that to improve software quality, we must learn from our past failures
- iv. Explain the concept of the CMM and CMMI process initiatives.

(b) Lecture 2: Software Development Processes

- i. Explain the difference between general software and real-time software.
- ii. Explain the relationship between Systems Engineering and Software Engineering
- iii. Define the concept of an Architecture
- iv. Compare and contrast the aspects of the Waterfall model, the V model, spiral models, and other software development models.
- v. Explain the ROPES Software Development Process (Rapid Object Oriented Process for Embedded Systems)

2. Week 2 - Requirements and Use Cases

(a) Lecture 1- Requirements

- i. Recognize the relationship between different types of requirements within the realm of software engineering
- ii. Compare and contrast operational requirements, quality of service requirements, parametric requirements, design, and implementation requirements.
- iii. Critique the wording of a requirement and constraints.

(b) Lecture 2 - Use Cases

- i. Define a use case
- ii. Interpret the meaning of a use case diagram.
- iii. Define Actor
- iv. Explain the relationship between Use Case Diagrams and Use Case Scenarios
- v. List the items present in a use case scenario
- vi. Construct a use case scenario for a given problem
- vii. Explain how the level of detail in use cases may change throughout the phases of the software development process.

3. Week 3

(a) Lecture 1- Introduction to Software Reviews

- i. Explain the concept of a software review
- ii. Illustrate the flow for a typical software review
- iii. Recognize the applicability of checklists to improving the review process
- iv. Identify the risks of reviewing at too fast or too slow of a rate.

(b) Lecture 2 - Introduction to Version Control and Configuration Management

- i. Explain the concept of a version control system
- ii. Recognize the importance of disciplined configuration management to the successful completion of a software development project
- iii. Understand the model of a client server configuration management system.
- iv. Define repository
- v. Define local working copy
- vi. Explain the concept of merging code files
- vii. Explain the capabilities of the GForge project management tool.
- viii. Explain the concept of labeling and branching
- ix. Explain how to interface with the GForge system using the TortoiseSVN client.
- x. Check in and check out source code and design documentation from an SVN repository.

4. Week 4 Object Domain Analysis

(a) Lecture 1 Object Domain Analysis

- i. Explain the purpose for Object Domain Analysis.
- ii. Explain the relationship between the use case model and the Object Model
- iii. Explain the mechanism used to connect object domain models with use case models
- iv. Apply key strategies for identifying objects within a problem domain
 - Noun Strategy
 - Services
 - Physical Devices
 - Persistent Information

(b) Lecture 2 Object Associations

- i. Explain the following object association strategies
 - Identify Messages
 - Identify Message Sources
 - Identify Message Storage Depots
 - Identify Message Handlers
 - Identify whole - part relations

5. Week 5 Defining Object Behavior

(a) Lecture 1 Defining Object Behavior

- i. Explain the relationship between simple, state, and continuous behaviors.
- ii. Define entry actions, exit actions, and activities.
- iii. Explain the transaction syntax for transitions.
- iv. Define guard condition.
- v. Define object state behavior using a UML state charts
- vi. Represent concurrent state machine behavior using Harel state machines.

(b) Lecture 2 Defining Object Interactions

- i. No class due to Good Friday.

(a) Lecture 1 Midterm Exam

- i. Pass the exam with an appropriate score.