

Swinburne University of Technology

School of science, Computing and Engineering Technologies

Contribution Document

Subject Code: SWE30003

Unit Title: Software Architectures and Design

Assignment number and title: 2. Object Design

Due date: 5th May 2024

Tutorial Day and Time: Wednesday 5:30 PM

Project Group: 05

Tutor: Mandeep Dhindsa

1. **Time spent** (*hours*): fill in the time spent on each significant work item in the following table.

Work Item	Tran Duc Anh Dang	Saw Ko Ko Oo	Luan Nguyen	Ertesam Naser Zarif
Abstract	✓ (0.5 hour)	✓ (0.5 hour)	✓ (0.5 hour)	✓ (0.5 hour)
Introduction	✓ (0.5 hour)			
System Requirements and Analysis	✓ (1 hour)	✓ (0.75 hour)	✓ (1 hour)	✓ (1 hour)
System Design Overview	✓ (2 hour)			
CRC Cards	✓ (2 hour)		✓ (2 hour)	
Design Heuristics and Patterns		✓ (1 hour)		✓ (1 hour)
Bootstrap and Initialization Process	✓ (1 hour)			(1 hour)
Design Verification and Scenarios		✓ (1 hour)	✓ (1 hour)	
Work Review	✓ (Daily Review)	✓ (Daily Review)	✓ (Daily Review)	✓ (Daily Review)

Note: Please adjust and add more rows, as necessary.

Description of contributions (each member provides a brief description of their contributions):

Tran Duc Anh Dang

- I was writing the abstract, contributing to the system requirements/analysis, system design overview, CRC cards, bootstrap and initialization and reviewing the work of other members. I have actively reviewed other works to make sure it aligns with the requirements.

Saw Ko Ko Oo

- I contribute to abstract, system requirements and analysis while also reviewing works from others.

Luan Nguyen


- I contribute to Abstract, System Requirements and Analysis, Design Heuristics and Patterns and Daily Work Review. I work with Anh face to face daily as we have multiple subjects together.

Ertesam Naser Zarif

- Wrote the abstract, worked as a team on voice chat for system requirements and analysis, design heuristics and patterns, bootstrap and initialization process. Reviewed other people part as well


... ..

2. Description and evidence of systematic collaboration / teamwork on each significant work item (eg, discussions, mutual revisions and reviews, etc.):

 **Zarif** 04/30/2024 7:38 PM
hi guys, based on the feedback from assignment 1, i think these are the requirements that we can use for assignment 2

- Reservation Management + Walk-in Management :Manage reservations and walk-in customers through the same system.
- Order Processing: Process orders from the customer and forward it to the kitchen.
- Payment Processing: Allow for multiple forms of payments for customers and create invoices.
- Inventory Management: Automatic management of inventory through items being used.
- Table Turnover Management: Maintain table turnover between departing and arriving customers in the system.
- Customer Feedback Collection: Provide a form for customer feedback for future review.
- Staff Scheduling: Create an interface for staff roster schedules. (edited)


its basically the tasks we had in assignment 1 that mandeep did not have a problem with. please let me know if you guys want to add / remove from here

 **anh dang** 04/30/2024 8:17 PM
yep, I'm working on the report tomorrow
are you down for a call tomorrow?


👍 1

I believe @Luan Nguyen and I will go over everything tomorrow though

😊 1

 **Zarif** 04/30/2024 8:25 PM
should be ok

May 1, 2024


 **Nathan Ko** 05/01/2024 9:08 PM
this is what I learnt in week6 about heuristics

- 1.Heuristics for System Classes
- 2.Heuristics for Inheritance
- 3.Heuristics for Abstract Classes
- 4.Heuristics for Base Classes

I read the HD sample but I just wanna know about
Design Heuristics and Patterns
Application of Design Heuristics: Specific heuristics used in the design.

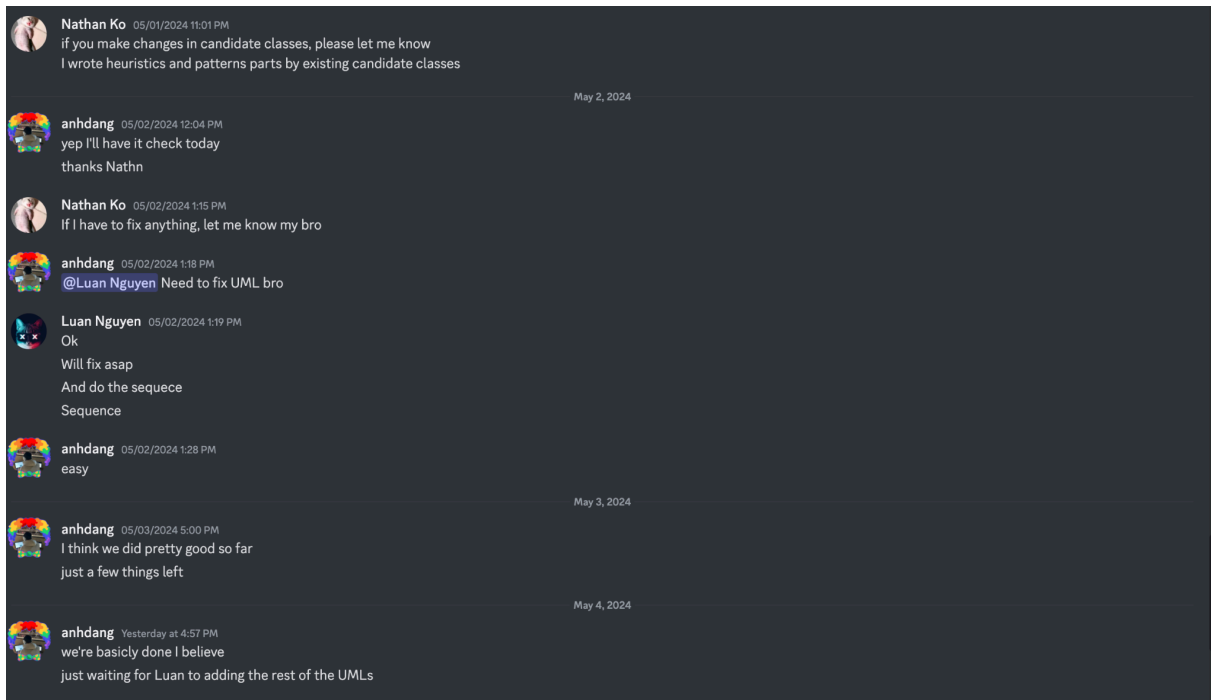
- 5.Heuristics for Associations

the answer in the HD sample are just copy of the definition of the heuristics definition from the lectures
I just wanna know that I need to follow HD sample or I need to expand it


 **Nathan Ko** 05/01/2024 9:53 PM
I wrote it according to the week 6 lecture note for design heuristic

- 1.Heuristics for System Classes
A class should capture one and only one key abstraction
Keep related data and behaviour in one place
Distribute system intelligence horizontally as uniformly as possible, that is, top-level classes of the system should share their work uniformly
Do not create god classes/objects in your system
- 2.Heuristics for Inheritance
It should be illegal for a derived class to override a base class method with a method that does nothing
Explicit case analysis on the type of an object is usually an error. The designer should use polymorphism in most cases
In practice, inheritance hierarchies should be no deeper than an average developer can keep in his/her short-term memory
In theory, inheritance hierarchies should be deep - the deeper, the better
Inheritance should be used only to model a specialization hierarchy
Derived classes must have knowledge of their base class (by definition), but base classes should not know anything about their derived classes
- 3.Heuristics for Abstract Classes
All abstract classes must be base classes
All base classes should be abstract classes
- 4.Heuristics for Base Classes
Factor the commonality of data, behaviour and/or interface as high as possible in the inheritance hierarchy

Interaction Heuristics
Minimize the number of classes with which another class collaborates
Minimize the amount of collaboration between a class and its



1. Signed by all members:

ID Number	Name	Signature
<u>103995439</u>	<u>Tran Duc Anh Dang</u>	<u>Anh</u>
<u>103834192</u>	<u>Ertesam Naser Zarif</u>	<u></u>
<u>104150310</u>	<u>Saw Ko Ko Oo</u>	<u>saw</u>
<u>103812143</u>	<u>Luan Nguyen</u>	<u>Luan</u>