Software System Design:

A Recipe for Success

4002-218

Paul Walter

Table of Contents

Before We Start 3

System Description: Fallout-land Messaging System 3

Create GUI Mockups 3

Textual Analysis: 3

Refine list 4

CRC (Class, Responsibility, Collaboration) Card Creation 4

State Diagram Creation 5

Class Diagram Creation 5

Sequence diagram creation (verifies your design) 6

Method documentation 6

Method implementation. 6

## Before We Start

Generally the projects you will be creating will be for a client. I would like to mention that you should never make unverified assumptions about the software system you are building. Always verify your findings with you client before you move on to the next phase of analysis, design, or implementation.

The following sections are listed in the order they should be attempted.

## System Description: Fallout-land Messaging System

Start out with a description of your system that either you, or your client writes. In this case we started with this Fallout-3 inspired software system description (to make it more dramatic):

“I need a messaging system that doesn’t use the Internet. A message has a recipient, a sender, and message text. A mailbox can store messages. Each user has at least 2: sent and received. The user needs to be able to log in, send messages to other users, read their own messages, and log out.”

## Create GUI Mockups

Once you have received the system description, it is time to create visual GUI mock-ups that the user can look up and ok (see the attached power point document). This will ensure that you don’t overlook or make any assumptions about your domain.

Our visual mockup-up helped us to discover that we were missing a verb (and consequently a state) in our description: “list messages. “

## Textual Analysis:

Pick out the nouns and verbs: Nouns become your software objects or attributes, verbs become methods and make an initial comment on what kind of element they might be: class, attribute, state or feature. Here is a first pass:

**Nouns**:

Messaging System //class

Internet //nope

Recipient //attribute of Message

Sender // attr of message

Message //class

Text // attr of message

Mailbox //class

Sent Received //state or subclass.

User // class

**Verbs**:

Log in //feature

Read // feature

Store // feature

Logout // feature

Send // feature

## Refine list

Take a second look at the refining list we just created in the previous step. We will want to refine this list more to something that is more useable: attributes are grouped according to what class they most likely fit into; features are evaluated to see if we keep them.

**Nouns**:

Messaging System //class

Message //class

* Recipient (User) //attribute of Message
* Sender (User) // attr of message
* Text // attr of message
* Date

Mailbox //class

* Sent Received //state or subclass.

User // class

**Features**:

Log in //keep

Read // keep

List messages // discovered in the visual mockups.

send //keep

Logout //keep

Send //keep

## CRC (Class, Responsibility, Collaboration) Card Creation

Next step is to create CRC cards. These will identify what responsibilities each class we have identified in the previous step should do, and what relationships exist with other classes in order to accomplish them.

Always view what responsibilities need to happen in terms of the features we identified in the textual analysis phase.

|  |  |
| --- | --- |
| Messaging System | |
| Send  Read Message  List Messages  Login  Logout | Message, User  Message, User  Message, User, Mailbox  User  User, Mailbox |

|  |  |
| --- | --- |
| Message | |
| Show text, recipient, sender, time | User, |

|  |  |
| --- | --- |
| Mailbox (sent or received) | |
| List messages | Message |

|  |  |
| --- | --- |
| User | |
| Verify username and password combo  Get mailboxes | MessageSystem |

## State Diagram Creation

After you have created CRC cards, it is time to identify what classes may have state, and to model them. In this example we identified two classes that had state:

1. MessageSystem (login, list, read, create)
2. Message(Sent/Unsent)

We asked our client, in this role-playing it was me (the instructor) and they said they didn’t care about Message state. In this situation we modeled the state for Message System

See supplied Power Point document for our State Diagram.

## Class Diagram Creation

Given the items we can find in our CRC cards we can create our Class Diagram (See included Power Point document).

To do this, consult the “Collaborator” section of the CRC card (the bottom right hand side of the card). These will be relationships we draw to those classes.

For example:

|  |  |
| --- | --- |
| Mailbox (sent or received) | |
| List messages | Message |

In the case of Mailbox, we draw a relationship between the Mailbox class and the Message class (just a straight line). Then we ask ourselves the criteria questions listed in “UML\_Guidelines\_And\_Examples”.

## Sequence diagram creation (verifies your design)

Then you can verify your design for each step by creating a Sequence Diagram for each feature that you identified (see included Power Point document).

This step is vital to ensure you don’t make unverified assumptions about your system and implement a faulty design.

Update your class diagram with any changes you discovered.

## Method documentation

Next create empty methods you discovered and verified in the preceding steps. Now document them as to what they are supposed to do.

## Method implementation.

Then implement it.