Heriot Watt University

ADVANCED SOFTWARE ENGINEERING

F21AS

Stage 2

Group Report

Jowita Knap	H00301565
Lorenzo James	H00309663
Lucía Parga Basanta	H00313273
Sabrina Chiesurin	H00314757

Contents

Heriot Watt University 1

Overview 3

Core functions 3

Limitations and Bugs 3

Extended functions 3

UML Class Diagram 4

Communication and Development using Agile Processes 5

Scrum 5

GitHub 5

Threads 6

Threads 6

Design Patterns 7

Singleton Pattern 7

Observer Pattern 7

MVC Pattern 7

Extensions 8

Stage 1 and stage 2 9

A comparison 9

Overview

In this second stage, the application is extended to simulate customers queuing at the counter and having their orders processed by coffee shop staff. We use threads and three design patterns that help us to develop the application. We use Singleton pattern to implement our log class, Observer and MVC patterns in our GUI. We use Scrum as agile framework. The application fully meets the specification.

Core functions

- The simulation consist of multiple serving staff and a single queue of customers that are waiting to be served. When new costumers arrive at the coffee shop, they join the back of the queue. When they reach the front of the queue, they will be processed by the next available member of serving staff;
- When it starts up, the application read in details of existing orders. These orders are gradually added to a queue;
- When a member of serving staff becomes free, they will begin to process the order at the front of the queue. Once the order has been processed, they move on to the next order in the queue;
- Once the queue is empty, the coffee shop closes, a report is generated and the program exits;
- In the GUI is it possible to see the customers and their orders waiting in the queue, and details of what each member of serving staff is currently doing throughout the simulation. The timing is slow enough to watch the displays changing;
- There is a log in which are kept records of the events as they happen. The log is written to a file when the application exits.

Limitations and Bugs

- **Test ???**
- **Too much If ???**
- **Java doc ???**

Extended functions

- Allow the user to press a button to alter the speed of simulation using runtime controls;
- Allow customers to order online in advance.
-

In this report are shown the details of the implementation and the methodologies used to develop and complete the work.

UML Class Diagram

Communication and Development using Agile Processes

Scrum

In this second stage of the work we use Scrum, a framework for effective team collaboration that help us to coordinate and split the works among the team members.

We had three weeks and we divided this time available into three Sprints of one week each. It was not possible to meet every day due to the timetables and the availability of the members. Anyway we decided to meet every Monday and Thursday to have “three-day scrum”, in which was possible to synchronize the activities and create a plan for the next three days.

During the first meeting, we divided the work in smaller tasks and we estimate the time needed to achieve them. To do so, we use Planning Poker estimation technique. This technique allows the members of the group to estimate the time needed for each task by playing numbered cards face-down to the table, instead of speaking them aloud. Once that the cards are revealed, we did some discussions about them. We decided to use this method to avoid being influenced by the opinion of the other members of the team.

At the end of each Sprint, we had a Sprint review. During this Sprint review, the work done was summarized and discussed. We then talked about the improvement to do in the next sprint to be more efficient.

((need to add that the last sprint was different and we have done what was left with peer review))

GitHub

We use GitHub to store the code, available at:
https://github.com/lorenzo456/AdvancedSoftwareEngineeringCoursework

Threads

In this section, we present the use of threads in the application.

Threads

Threads have been used in

Design Patterns

In this section, we present the use of design patterns in the application.

Singleton Pattern

Observer Pattern

MVC Pattern

Extensions

Stage 1 and stage 2

A comparison